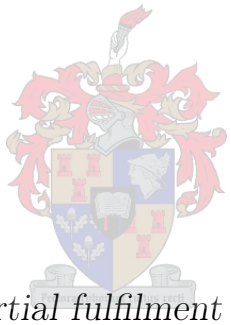
Development of a Thermal Analysis Tool for Superconducting Circuits

by

Bernard Hermann Venter



Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. Coenrad J. Fourie

March 2021

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2020/11/06 .....

Abstract

Development of a Thermal Analysis Tool for Superconducting Circuits

BH. Venter

Thesis: MEng (EE)

March 2021

Unforeseen heat generation or propagation adversely affects superconducting integrated circuits. The excess heat interferes with the performance of circuit elements sensitive to temperature, such as the Josephson junction (JJ). Research shows the increasing density of circuit elements due to the growing integration scale of superconducting circuits. The study aims to create a thermal analysis tool to simulate the heat propagation through a superconductor circuit before fabrication.

The tool was created by deriving the thermal conductivity of superconducting metals, developing a finite element model of the structure, and simulating the heat propagation iteratively for a set of test conditions. The primary factors limiting the operation of the simulation is the assumption of zero magnetic field and perfect thermal contact.

The heat transfer simulation through a Josephson Transmission Line (JTL) circuit demonstrated the operation of the thermal analysis tool. The thermal analysis performed on the JTL examined the effect an increasing bias current had on the surrounding area. The simulation showed a minimal temperature increase in the Josephson junction due to heat dissipated by the bias resistor. The large temperature increase was localised around the bias resistor. The thermal analysis tool successfully simulated the heat dissipation through a superconducting circuit for different test cases.

Uittreksel

Ontwikkeling van 'n Instrument vir Termiese Analise vir Supergeleidende Stroombane

("Development of a Thermal Analysis Tool for Superconducting Circuits")

BH. Venter

Tesis: MIng (EE)

March 2021

Onvoorsiene opwekking of verspreiding van hitte beïnvloed die supergeleidende geïntegreerde stroombane nadelig. Die oormatige hitte belemmer die werking van stroombaan elemente wat temperatuur sensitief is soos die Josephson junction (JJ). Navorsing toon aan die toenemende digtheid van stroombaan elemente as gevolg van die groeiende integrasie skaal van super geleidende stroombane. Die studie het ten doel om 'n instrument vir termiese analise te skep om die hitte-verspreiding deur 'n super geleier stroombaan te simuleer voordat dit vervaardig word.

Die instrument is geskep deur die termiese geleidingsvermoë van super geleidende metale af te lei, 'n eindige element model van die struktuur te ontwikkel en die hitte-voortplanting iteratief te simuleer vir 'n stel toets omstandighede. Die primêre faktore wat die werking van die simulasie beperk, is die aanname van nul-magnetiese veld en perfekte termiese kontak.

Die hitte-oordrag-simulasie deur 'n Josephson Transmission Line (JTL) stroombaan het die werking van die termiese analise instrument gedemonstreer. Die termiese analise wat op die JTL uitgevoer is, het die effek van 'n toenemende vooroordeel stroom op die omgewing ondersoek. Die simulasie het 'n minimale temperatuurtoename in die Josephson junction getoon as gevolg van hitte wat deur die vooroordeel resistor versprei word. Die groot temperatuurverhoging is gelokaliseer rondom die vooroordeel resistor. Die termiese analise instrument het die hitte-afvoer suksesvol gesimuleer deur 'n super geleidende stroombaan vir verskillende toetsgevalle.

Acknowledgements

I would like to acknowledge and thank:

- My supervisor, Prof Coenrad Fourie, for his guidance and support given throughout the project.
- Heinrich Herbst, Paul le Roux, Dr.Kyle Jackman, Dr. Joey Delport and Lieze Schindler for their valuable input and discussions.
- My parent for their moral support, encouragement,and prayers that got me where I am now.
- My flatmate for the encouragement and banter.
- My Lord and Saviour Jesus Christ, through which none of this would have been possible.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AMM	Acoustic Mismatch Model
CMP	Chemical Mechanical Planarization
CLI	Command-Line Interface
DC	Direct Current
DMM	Diffuse Mismatch Model
FEA	Finite Element Analysis
FEM	Finite Element Method
HDF5	Hierarchical Data Format version 5
HSTP	High-Speed Standard Process
IC	Integrated Circuit
JJ	Josephson Junction
JTL	Josephson Transmission Line
MIT-LL	Massachusetts Institute of Technology Lincoln Laboratory
PECVD	Plasma-enhanced chemical vapour deposition
PDE	Partial Differential Equation
RF	Radio-frequency
RRR	Residual Resistivity Ratio
SCE	Superconducting electronics
SFQ	Single-flux-quantum
SSR	Standard Sheet Resistance

Tc	Critical Temperature
VLSI	Very Large-Scale Integration
XDMF	eXtensible Data Model and Format
Nb	Niobium
Mo	Molybdenum
He	Helium
Si	Silicon

Variables

ρ	Density	$[\mathrm{kg{\cdot}m^3}]$
c_p	Specific Heat	$[\mathrm{J/kg{\cdot}K}]$
q	Rate of Heat Generation	$[\mathrm{W/m^3}]$

Subscripts

c	Critical condition
l	Liquid
s	surface
sat	saturated conditions
$surr$	surrounding environment
v	vapour or gas

Chapter 1

Introduction

1.1 Background

Recent improvements in the fabrication process of superconducting electronics (SCE), highlight the growing importance of understanding heat transfer in SCE. The advances made in the fabrication process, support progress made toward the Very Large-Scale Integration (VLSI) of Single Flux Quantum (SFQ) circuitry. The improvements in the fabrication process at Massachusetts Institute of Technology Lincoln Laboratory (MIT LL), show that there is an increase in the number of superconducting layers, and a decrease in the minimum layer linewidth [2–6]. These improvements promote the more compact placement of components on the wafer. Utilising the current SFQ5ee process [2] from MIT LL, complex superconductor circuits with nearly one million JJs and a circuit density of over 1.33 x 10^6 JJs per cm^2 was demonstrated [7, 8]. The recent SC1 process [6] nearly triples the JJ device count and critical current density with plans on increasing the device count to ten million JJs per chip [9].

With the increased number of circuit elements fabricated more densely, the heat spread through the superconductor circuit could adversely affect its operation. The greater circuit density might result in the placements of the bias resistor closer to temperature sensitive circuit elements, such as JJs. The resistor placement could contribute to an increase in temperature of the JJ, as the heat dissipated by the bias resistor increases. The amount of heat generated by the resistor depends on the bias current applied to the circuit. The bias resistor generates heat and induces a magnetic field from the applied bias current. The heat dissipated by the thin-film bias resistor influences the operation of the JJs and the subsequent SFQ circuit [4]. The increase in temperature influences the change in parameters such as jitter, SFQ pulse delay and the change in critical current due to thermal noise [10–12]. The heat dissipated by a bias resistor increases the temperature of the surrounding environment, thus reducing the critical current of the surrounding JJs and

superconducting wiring layers. In an extreme case, the excess heat could turn the superconducting elements back into its normal non-superconductive state [13].

1.2 Problem Statement

An undesirable increase in temperature has a negative impact on the operation of superconducting circuits. Potential heating issues can sometimes be overlooked during the design phase and only become apparent during the testing phase after fabrication. Accurate simulations and modelling of the superconductor circuits can help to identify potential problems before fabrication. This thesis aims to create a thermal analysis tool for superconductor circuits.

1.3 Objectives and Goal

The main goal of this thesis is to develop a tool to simulate the heat propagation through superconductor circuits. The main goal was divided into the following objectives:

- Investigate the thermal conductivity of superconductor metals at low temperatures and the effect temperature has on it.
- Derive a thermal load for the applied superconductor circuits.
- Develop a software tool for the numerical thermal analysis.
- Perform finite element analysis on the meshed superconductor
- Perform thermal analysis on one-, two-,and three-dimensional superconductor structures.
- Analyse the results obtained from the simulations and the influence temperature has on the surroundings.

1.4 Chapter Overview

The section provides an overview of the chapter structure of the thesis. The thesis is structured as follows: Chapter 2 discusses the literature relevant to the project. The chapter commences with an introduction to superconductivity. This is followed by the thermal conductivity in superconducting circuits and close with the thermal analysis approach discussing the heat transfer and software solutions required to solve the simulations. Chapter 3 is devoted to the derivation of the thermal conductivity of superconducting metals at low temperatures. The thermal conductivity of some of the materials present in the

superconductor circuit was also identified. Chapter 4 derives the thermal load and numerical calculations for the thermal simulation. The weak form of the heat transfer equation, boundary conditions and applicable conditions are calculated. Chapter 5 explains the development of the software component for the thermal analysis model. The results from the thermal simulation for one-, two- and three dimensions are demonstrated in Chapter 6. Finally, Chapter 7 provides an important conclusion and provide recommendations for future work on the subject.

Chapter 2

Literature Review

2.1 Introduction

The phenomenon where the electrical resistance of various solids reduces to zero as the temperature decreases below a critical temperature is called superconductivity. The materials exhibiting these effects are known as superconductors [14]. This chapter provides a review of the literature related to superconductors and the thermal analysis approach. The introduction to the basics of superconductivity is supplied, including the different superconductor models. The heat conduction of Type II systems is then discussed. The focus of the review then shifts to the thermal analysis approach. The mechanisms of the heat transfer process and simulation software solutions are mentioned.

2.2 Superconductivity Background

The superconductivity phenomenon was first observed in 1911 by the physicist Heike Kamerlingh Onnes and his team [14]. The discovery came after investigating the electrical resistance of mercury at cryogenic temperatures. They observed the electrical resistance of the mercury suddenly disappeared below 4.2K. The mercury entered the superconducting state. Heike Kamerlingh Onnes went on to win the 1913 Nobel Prize in physics for his low-temperature research [15].

2.2.1 Basic properties

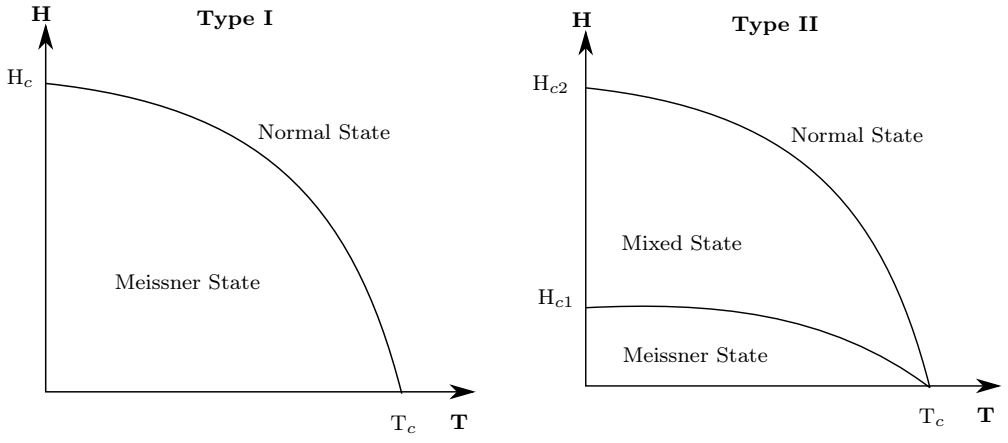
During the transition to the superconducting state, the magnetic flux applied to the superconductor is repelled. The expulsion of the magnetic field is known as the Meissner effect [14]. The magnetic field induces a current in the superconductor structure. The induced current produces a magnetic field equal in magnitude but opposite in direction to the applied magnetic field. The resulting magnetic field expels the applied magnetic field from the supercon-

ductor. Since the electrical resistance is zero, the induced current flows for as long as a magnetic field is applied to the structure [14].

The material can lose its superconductivity when subjected to large currents, magnetic fields or high temperatures. The maximum magnetic field or current density that can be applied, for T = 0, is the critical magnetic field, (H_c), and critical current density (J_c) respectively [16]. The critical magnetic field and current density decrease with an increase in temperature. Superconductors are cooled below $0.6T_c$ for most engineering applications.

2.2.2 Type I and II superconductors

The superconductor is classified as Type I or Type II, depending on how it reacts to the applied magnetic field. Type I superconductors expel all magnetic flux from the superconductor for $H < H_c$. In this region, the superconductor is in the Meissner state and is perfect diamagnetic. The superconductor enters the normal state when the magnetic field increases past the critical magnetic field value. Type II superconductors contain an intermediate state known as the mixed or vortex state [17]. A Type II superconductor exhibits a lower and higher critical magnetic field H_{c1}, and H_{c2} respectively. For $H < H_{c1}$, the superconductor is perfect diamagnetic just like Type I superconductors. As the magnetic field increases, the superconductor enters the mixed state. For $H_{c1} < H < H_{c2}$, a finite amount of magnetic flux penetrates the superconductor and is no longer perfect diamagnetic [18, 19]. Figure 2.1 shows the phase diagram for Type I and Type II superconductors.



(a) Type I superconductor.

(b) Type II superconductor.

Figure 2.1: The magnetic field and temperature phase diagram for Type I (left) and Type II (right) superconductors.

The upper critical magnetic field of Type II superconductors are much greater in magnitude than the lower critical magnetic field. The lower critical field is in the same range as the critical field of Type I superconductors. Since

Type II superconductors can handle much greater magnetic fields, it is the preferable superconductor used in fabrication. Type I superconductors are usually pure metals including mercury and lead. The Type II superconductor family includes niobium, alloys and complex oxides of ceramics, to name a few [14].

2.2.3 Superconductor models

The development of a heat transfer model for superconductors requires a good understanding of superconductivity. Multiple different superconductivity models have been developed over the years to help explain the fundamental physics of superconductivity. Two of the models developed are BCS Theory and the Classical Model of Superconductivity Each model has its advantages and limitations in explaining the basic physics of superconductors.

2.2.3.1 BCS Theory

The BCS theory [20, 21] is a comprehensive microscopic theory that describes the origins of superconductivity. The BCS theory was developed by John Bardeen, Leon Cooper, and Robert Schreiffer in 1957. The name of the microscopic theory of superconductivity was derived from the first character of each researcher's name. The theory can microscopically describe the properties of conventional low-temperature superconductors. For high-temperature superconductors ($T_c > 77K$), the BCS theory becomes inadequate [14].

The BCS theory describes the operation of superconductors by condensing the electrons that carry lossy current into Cooper Pairs. The Cooper pairs (also called *superelectrons*) are formed by the weak attractive forces between two electrons due to the electron-phonon interaction at low temperatures. The binding energy keeping the Cooper pairs together is $2\Delta(0)$, where $\Delta(0)$ is the gap energy at $T = 0K$. The electrons are bonded together until the phonon energy exceeds $2\Delta(0)$. At that point, the electrons push each other apart and return to the normal electron state [19].

The weak binding force keeping the electrons together is highly susceptible to a temperature increase. The amount of Cooper pairs present is dependent on the temperature. For temperatures near T_c, the gap energy and the number of Cooper pairs vary greatly with temperature. As the temperature decreases, more normal electrons condense into Cooper pairs [18, 19]. For $T > 0K$, the decrease in normal electrons due to the formation of Cooper pairs follow the exponential function $\exp(-\Delta(T)/k_BT)$. At extremely low temperatures, all the normal electrons are Cooper pairs, and the gap energy is temperature independent. The formation of Cooper pairs decreases the heat transferred due to the scattering of normal electrons.

2.2.3.2 Classical Model of Superconductivity

The classical model of superconductivity describes the perfect diamagnetism and zero resistance properties of superconductor materials [14]. The classical model describes how the superconductor material functions but cannot explain why the material functions as it does. The London equations, derived by Fritz and Heinz London in 1935 [22], contains the two material properties mentioned above. The first London equation describes the electron motion phenomenon in a perfect conductor. The second London equation incorporates the Meissner effect into the first equation. The first and second London equation state:

$$\mathbf{E} = \frac{\partial}{\partial t}(\Lambda \mathbf{J}). \tag{2.1}$$

and

$$\nabla \times (\Lambda \mathbf{J}) = -\mathbf{B}. \tag{2.2}$$

where $\mathbf{E}$ is the electric field, $\mathbf{J}$ is the current density, $\mathbf{B}$ is the magnetic field and Λ represents the superconducting properties of the material. Λ is defined as:

$$\Lambda = \frac{m^*}{n^*(q^*)^2}. \tag{2.3}$$

where m^*, n^* and q^* are superelectron mass, superelectron density and superelectron charge, respectively. Assuming the material is uniform and nondispersive, the magnetic field is represented as:

$$\left(\frac{\mu_0}{\Lambda} - \nabla^2\right)\frac{\partial}{\partial t}\mathbf{H} = 0. \tag{2.4}$$

where

$$\lambda = \sqrt{\frac{\Lambda}{\mu_0}}, \tag{2.5}$$

where μ_0 is the permeability of free space and λ is the London penetration depth of the superconductor. The London penetration depth is the distance the external flux penetrates the superconductor without loss of superconductivity properties [14].

The classical model does not mention the thermodynamic process of superconductors up to this point. Superconductor materials undergo a thermodynamic change in state. As the temperature decreases below T_c, the material transitions from the normal to superconductive state. The thermal properties

are added to the classical model by making use of the two-fluid model. The two-fluid model was first proposed by Gorter and Casimir [23] in 1934. The two-fluid model explains the zero DC resistance for superconductor materials in the temperature region $T < T_c$ due to the formation of superelectrons.

The model assumes the total current density through the superconductor is the sum of superelectrons and electron current densities. At 0K, all the current carriers are superelectrons. At $T > T_c$, the current carries are normal electrons. For the intermediate region, the current carriers are a combination of normal electrons and superelectrons [14]. Throughout the region, $T_0 < T < T_c$, the superelectrons increase, and the normal electrons decrease with an increase in temperature. Due to the change in current density with temperature, the temperature-dependent London penetration depth is as follows:

$$\lambda(T) = \frac{\lambda_0}{\sqrt{1 - (\frac{T}{T_c})^4}} \quad \text{for} \quad T \leq T_c. \tag{2.6}$$

where λ_0 is the London penetration depth at 0K. Combining (2.3) and (2.6), the temperature-dependent Λ is derived as:

$$\Lambda(T) = \frac{m^*}{n^*(q^*)^2}\left(\frac{1}{1 - (\frac{T}{T_c})^4}\right) \quad \text{for} \quad T \leq T_c. \tag{2.7}$$

The first and second London equations incorporating the thermodynamic properties of superconductors are thus stated as:

$$\mathbf{E} = \frac{\partial}{\partial t}(\Lambda(T)\mathbf{J}_s) \quad \text{for} \quad T \leq T_c. \tag{2.8}$$

and

$$\nabla \times (\Lambda(T)\mathbf{J}_s) = -\mathbf{B} \quad \text{for} \quad T \leq T_c, \tag{2.9}$$

where $\mathbf{J}_s$ is the superelecton current density. The classical model provides a good approximation of what happens to superconductor materials cooled below T_c. The classical model describes bulk superconductivity with ease, but struggles with the more complex Josephson quantum effects. The classical model was able to describe the zero DC resistance, Meissner effect and temperature influence on the superconductor material. The model is limited in the sense that it cannot describe the interaction between the three effects mentioned above [14].

2.3 Type II Superconductor Thermal Conductivity

The thermal conductivity measures a material's ability to conduct heat [24]. The derivation of thermal conductivity through superconductors has been investigated through multiple theories and experiments [21, 25–30]. The thermal conductivity is measured in both the superconductive and normal state. The normal state consists of two factors, the electron and the phonon contribution stated as:

$$k = k_e + k_g \tag{2.10}$$

where k_e and k_g is the electron and phonon thermal conductivity, respectively. For superconducting metals in the normal state, the electron contribution is dominant. The electron-phonon and lattice imperfection scattering limit the electronic contribution [31]. The two-fluid model helps predict what happens to the thermal conductivity in the superconductor state as the temperature decreases past T_c [32]. In the superconducting state, the electronic contribution decreases with temperature as electrons condense into Cooper pairs. The formation of Cooper pairs reduces the heat energy transfer in the superconductor since Cooper pairs do not contribute to entropy transport.

As the electron contribution decreases, the phonon contribution becomes more dominant. The phonon component becomes dominant at temperatures below 0.2 T_c to 0.3 T_c due to the disappearance of the electronic component and the increase in phonon free path [21]. The thermal conductivity at exceptionally low temperatures is due to the scattering of phonons by crystal boundaries [33]. The phonon and electron contribution to the thermal conductivity in the superconductor state was calculated by Bardeen, Rickayzen, and Tewordt [21] using the BCS theory. The gap energy component of the electron contribution of thermal conductivity in the superconductor state depends on the change in temperature and magnetic field [21].

The impurities in the superconductor influence the thermal conductivity of Type II systems. Pure ($1 >> \xi_0$) and dirty ($1 < \xi_0$) superconductors have different coherent lengths depending on the number of impurities [14]. The purity depends on the dominant scattering factor in a Type II superconductor. The dominant scattering factor is phonons for pure and impurities for dirty Type II superconductors [29]. The research focuses on the pure limit.

The thermal conductivity of Type II superconductors is not only influenced by temperature, but also the magnetic field. A Type II superconductor enters the mixed state when placed in a sufficiently large magnetic field. The thermal conductivity in the mixed state depends on the size and direction of the magnetic field with respect to the heat flow through the superconductor [29]. For high magnetic fields in the region close to H_{c2}, Maki [28] determined the

electronic contribution of the thermal conductivity for the superconducting region. Maki [28] showed that the thermal conductivity decreases as $(H_{c2}-H)^{1/2}$ with an infinite slope. The thermal conductivity in the mixed state is highly anisotropic. At temperatures near T_c, the thermal conductivity parallel to the applied magnetic field is greater than the thermal conductivity perpendicular to the magnetic field. The anisotropy of the thermal conductivity is inverted at the low temperature range. At low fields, the effects on the thermal conductivity are minimal [29].

2.4 Thermal Analysis Approach

The thermal analysis process measures the effect that temperature change has on the material properties and the heat flow of the system. The change in material properties of the system influences the overall performance of the system. The thermal response through the system is analysed by performing a thermal simulation. Thermal simulation enables the user to investigate the influence of temperature on structures that are difficult to measure, such as the inside layered structures. The accuracy of the thermal analysis simulations depends on understanding the heat transfer mechanisms, thermal properties of the materials and applicable boundary conditions. The following sections discuss the different heat transfer mechanisms, the numerical method for heat transfer and the thermal analysis software packages that are currently available.

2.4.1 Heat Transfer

Heat transfer is the rate of energy transferred from one system to another due to a difference in temperature [24]. The temperature difference between the systems are the driving force for the heat transfer process and it stops once equilibrium between the systems are achieved. The heat transfer process depends on the laws of thermodynamics for its supporting structure [34]. The first law of thermodynamics is known as the preservation of energy principles. The law states that the change in the internal energy of a system is equal to the energy entering through the boundary and the heat generated in the system. The mechanisms contributing to heat transfer include conduction, convection and radiation [24, 34].

2.4.1.1 Conduction

Conduction is the transfer of energy due to particle collision. Heat is transferred from particles with more kinetic energy to surrounding lower energetic particles in the body. The transfer of energy continues until the body reaches an equilibrium state. Conduction transfers energy through contact with another medium and is the most common heat transfer mechanism. Fourier's law

of thermal conduction dictates the rate of heat transfer due to conduction. The heat conduction rate depends on the temperature gradient, thickness, cross-sectional area, and the physical properties of the material used. *Fourier's law of thermal conduction* [24] states:

$$\dot{Q} = -kA\frac{dT}{\partial x}. \tag{2.11}$$

where $\dot{Q}$ is the rate of heat conduction per unit time (W/s), k is the thermal conductivity (W/m K), A is the cross-sectional area (m^2), dT is temperature difference and dx is the thickness.

2.4.1.2 Convection

Convection is a form of heat transfer from a surface to an adjacent moving fluid medium. The fluid media can either be a liquid, gas, or vapour. Convection is heat conduction with an added fluid motion parameter. Convection has two forms: forced convection and natural ("free") convection [34, 35]. Forced convection occurs when the fluid is forced across the surface by means of an external force such as a pump. Natural convection makes use of buoyancy forces and differences in densities of the fluid. The heated fluid rises away from the surface and is replaced by cooler fluid. The heat is thus transferred from the surface to the surrounding liquid.

The change of phase during the heat transfer process is also considered part of convection. Fluid motion is involved in the process of vaporisation as the liquid is transformed into gas [24]. The rate of heat transfer due to conduction is complex due to the problem of using conduction with fluid motion. The more turbulent the fluid, the more efficient the heat transfer. The convection heat transfer coefficient is not only dependent on the material, but also on the surface of the material. Multiple factors influence the coefficient and is usually measured in the laboratory and is not inherent to the material. The simplified rate of convection heat transfer can be represented as *Newton's Law of cooling* [24]. The following equation denotes the heat transfer from the surface to the surrounding liquid medium.

$$\dot{Q} = hA(T_s - T_\infty), \tag{2.12}$$

where h is the convection heat transfer coefficient in (W/m^2 K), T_s is the surface temperature and T_∞ is the fluid temperature sufficiently far away from the surface.

2.4.1.3 Radiation

Thermal radiation is the transfer of heat due to electromagnetic waves emitted from heated bodies. The energy released in the form of electromagnetic waves

is due to particle collisions and vibrations. The particle collisions and vibrations depend on the internal energy of the body. The energy emitted from the heated surface propagates to cooler surfaces, where it is absorbed and turned into heat. Thermal radiation does not require a medium to propagate through and is the only heat transfer mode able to propagate through a vacuum. All bodies above zero Kelvin radiate heat to its surroundings. Thermal radiation only becomes dominant at very high temperatures. Thermal radiation depends on the surface from which it radiates and is absorbed [24, 35]. For ideal cases, all the energy from the radiation is absorbed or emitted from a surface. Such ideal cases are called black body surfaces. The rate due to thermal radiation is calculated by making use of *Stefan-Boltzmann law* [24, 34, 35] as follows:

$$\dot{Q}_{rad} = \varepsilon \sigma A_s ({T_s}^4 - {T_{surr}}^4), \tag{2.13}$$

where ε is the emissivity of the surface, σ is the Stefan-Boltzmann constant and T_{surr} is the temperature of a surface surrounding the body. The emissivity of the surface is in the range of $0 \leq \varepsilon \leq 1$, where $\varepsilon = 1$ is perfect emissivity.

2.4.2 Thermal Contact Conductance

Superconductor circuits are multilayer structures composed of superconductive, normal metal, anodized, and dielectric layers [2–6, 36]. The heat transfer through layers of dissimilar materials in thermal contact is known as thermal contact conductance [24, 37]. The layers are in thermal contact when heat energy exchanges between the layers. The heat conduction depends on the interface and thermal contact conditions between the layers. The layers are said to make perfect thermal contact if the interface between the layers is perfectly smooth and no temperature drop is present [24].

The interface between layered structures is never perfectly smooth. Even the smoothest surface has many flaws at the interface when checked microscopically. The defects appear as peaks and valleys when viewed under a microscope. The imperfections between the layers decrease the total thermal contact area, resulting in a temperature drop at the interface. The peaks provide contact spots between the layers and the valleys leave gaps between the layers. When applying heat flux to the layered structure, the heat flux restricts down to the solid-solid contact spots. The heat transfer through the gaps are exceedingly small (if a fluid is present) or neglected (if in a vacuum) [24, 38, 39]. The heat flux through the interface is derived using the equation derived from *Newtons Law of Cooling* [24] as:

$$\dot{Q}_{interface} = h_c A \Delta T_{interface}, \tag{2.14}$$

where h_c is the thermal contact conductance coefficient (W K /m^2), A is the

apparent interface area (m^2), and $\Delta T_{interface}$ is the temperature difference at the interface. The h_c coefficient is similar to the term used in the convection heat transfer mechanism. The thermal contact resistance is the inverse of the thermal contact conductance expressed as:

$$R_c = \frac{1}{h_c}, \tag{2.15}$$

where R_c is the thermal contact resistance (m^2 K / W). The thermal contact resistance concept provides a better explanation for the effects the interface has on heat transfer. Several factors affect thermal contact resistance that includes surface roughness, material properties, temperature, and pressure at the interface. [24,37,39,40]. For materials at cryogenic temperatures, the thermal boundary resistance calculations become more complex. The factors that affect the thermal resistance at cryogenic temperatures include the Kapitza resistance between the circuit and liquid He [41, 42], the thermal boundary between normal and superconducting layers [19, 43], and the thermal conductivity of the layers. The thermal boundary resistance at the interface can be predicted using the Acoustic Mismatch Model (AMM) and the Diffuse Mismatch Model (DMM) [40, 42] at exceptionally low temperatures. Predicting the temperature drop between in the interface region is not always possible and has to be measured experimentally [37]. This increases the difficulty in simulating the heat transfer through structures, especially in the nanometer (nm) range.

2.4.3 Numerical Heat Transfer

Both analytical and numerical methods can calculate the heat transfer through a structure. Finding the analytical solution for heat transfer problems with complex geometries, temperature-dependent properties and boundary conditions, is not always possible. Solving the heat transfer problem using numerical methods provides a close approximation of the temperature distribution through the system. The complex heat transfer problem is solved by making use of the Finite Element Analysis (FEA) numerical method [44]. The thermal analysis of the structure, using FEA, models the conductive heat transfer based on the thermal conductivity of the material. The boundary conditions applied to the problem approximates the heat transfer due to convection and radiation. The approximate heat transfer due to convection and radiation is sufficient for this case. The heat transfer problem is structured mathematically using Partial Differential Equations (PDEs). The nonlinear PDE describing the heat transfer through the structure is solved using the Finite Element Method (FEM) [45, 46].

The FEM subdivides the domain of the complex problem into several smaller finite elements for easier calculation. The subdivision of the domain

helps with the accurate representation of complex geometries [47]. The unknown value over each element is approximated using the known values of the applied function. The heat transfer problem with approximate boundary conditions is solved over each finite element. The results over each finite element are combined ("assembled") over the entire domain of the problem to find the final solution to the problem [46]. The FEM problem can accurately be solved by making use of thermal analysis software to simulate the thermal model.

2.4.4 Thermal Analysis Software

The following section discusses the available open-source finite element software packages available for heat transfer problems. The list contains some of the most popular tools currently available. It consists of libraries and software packages.

2.4.4.1 FEniCS

FEniCS is an open-source collection of software packages with a C++ and Python interface designed to solve PDE problems. FEniCS was originally developed in 2003 through international collaboration [47]. The current version, 2019.1.0, was released on 19 April 2019 and is available on the FEniCS Project website. FEniCS is designed around the Ubuntu environment and is available on Windows, OS X and Linux through a high-performance customisable docker file. FEniCS provides clear and informative documentation, this includes multiple books and tutorials, [47, 48].

The FEniCS Project is a software solution to automate the PDE solving process [49]. FEniCS converts the scientific model into a more manageable finite element method. FEniCS provides a convenient built-in mesh function using *mshr* and allows imported mesh data using their XDMF format [47]. The variational form of the PDE under investigation together with the mesh data and relevant boundary conditions, are used as input for the solver. FEniCS supports multiple Finite Element solvers and preconditioners to optimally solve the presented problem [49]. FEniCS allows for identification of boundaries and subdomains. FEniCS is easy to get started and provides powerful parallel computation capabilities for more complex problems. FEniCS allows for the thermal and electromagnetic solver, flux calculations and iterative solvers.

2.4.4.2 MFEM

MFEM is an open source, lightweight C++ library for solving PDE using FEM [50]. MFEM provides the functions required to build a finite element algorithm for problems relating to electromagnetics, elasticity, definition, fluid mechanics etc. MFEM was initially released on July 2010 with version 4.1 released on 10 March 2020. Version 4.1 is available on GitHub, developed and

maintained by Lawrence Livermore National Laboratory [51]. MFEM is easy to build on Windows, Linux and OS X from source. MFEM provides good examples and documentation for getting started.

MFEM functions as a lightweight Finite element toolbox for assisting in the development of finite element algorithms for the solution of PDE problems. MFEM then transforms the finite element function into its linear algebra vector and sparse matrix form [50]. MFEM supports multiple finite element spaces and finite element solvers. Just like FEniCS, MFEM supports multiple mesh types and is updated regularly. A Python wrapper has been written for MFEM [51]. MFEM supports a large number of formats.

2.4.4.3 FreeFEM++

FreeFEM is an open-source integrated software package for solving partial differential equations. FreeFEM makes use of a high-level programming language unique to the FreeFEM software solution. The language is a C++ idiom and is more comparable to LaTeX. FreeFEM was initially created using Pascal in 1987 and was later rewritten into C++ [52]. The current package is version 4.4 and is available on FreeFEM's download page or installed using the source from their GitHub page [52]. The software is compatible with macOS, Windows and Linux.

The language used requires a learning period to become familiar with the syntax used. FreeFEM makes use of the variational form of the PDE to solve the equation using FEM. FreeFEM supports multiple finite element spaces and finite element solvers for 2D and 3D problems. FreeFEM only supports the 2D waveguide boundary conditions in Maxwell equations. The software is well documented with relevant examples and tutorials to learn their language. Matlab or Octave is used to view the plotted data and the mesh is saved using the .msh format. FreeFEM makes use of parallel processing using MPI.

2.4.4.4 Deal.II

Differential Equations Analysis Library II (deal.II) was initially released in 2000 as a C++ library to solve PDE using an adaptive finite element method [53]. Version 9.2.0 was released on 20 May 2020 and is available to download the prebuild packages or via source code on GitHub [54]. Deal.II works on Windows, Linux and macOS via source or a docker container [53]. The software is properly documented with extensive tutorial programs available to help new users.

Deal.II provides an easy to use tool to quickly set up a finite element solver with adaptive meshing and degrees of freedom handling. The software supports a wide variety of different finite elements, including Lagrange and Nedelec elements, to name a few. The Deal.II software package acts as a standalone linear algebra library with the ability to interface with other libraries to

increase functionality. The software is highly portable and supports a variety of compilers.

2.5 Summary

This chapter provides a broad introduction to literature applicable to this project. The chapter started with an introduction to superconductivity and a description of the models of superconductivity. The influence of temperature and magnetic fields on the thermal conductivity of Type II superconductors was discussed for both the purely superconductive and mixed state. It was found that for pure superconductors the dominant scattering factor contributing to the thermal conductivity was due to phonons. The thermal analysis approach was mentioned with the different heat transfer mechanisms. The thermal contact conductance discussed the interface conditions between layered structures and the factors that influence the heat conduction between layers. The chapter closes with a discussion of the different FEM software packages and highlights the importance of careful heat management for temperature-sensitive components.

Chapter 3

Low Temperature Thermal Conductivity

Thermal conductivity describes the ability of a material to conduct heat. The increase in the thermal conductivity of the material increases the rate of heat transfer. Due to this, the thermal conductivity influences the stability of the material at cryogenic temperatures [37]. The main heat carriers for metals are electrons and lattice vibrations. The heat carriers are additive, and the total thermal conductivity can be expressed as:

$$k = k_e + k_g, \tag{3.1}$$

where k_e and k_g is the thermal conductivity due to free electron motion and lattice vibrational waves (or phonons) respectively. The thermal conductivity equation can be used for both the normal and superconducting state. The heat conduction depends on the material and how the electrons and phonons interact.

3.1 Electron Thermal Conductivity

Electron thermal conductivity is the dominant contributing factor in pure metals even though the lattice vibration contribution is more prominent. The dominance stems from the velocity of the particles. The Fermi velocity of the electrons is faster than the velocity of sound of the phonons, resulting in the dominance [35, 37]. The electron thermal conductivity is calculated in both the normal and superconducting region.

3.1.1 Normal Region

For $T > T_c$, the electron thermal conductivity is in the normal state, k_{en}. The flow of electrons in the normal state is hampered by two scattering processes.

The first is electron scattering due to phonons, k_{eng}. The second is electron scattering due to impurities and defects in the metal, k_{eni}. The total thermal resistance to the flow of electrons is found by adding the two scattering terms such that [31, 55]:

$$\begin{aligned} W_{en} &= \frac{1}{k_{en}} \\ &= \frac{1}{k_{eng}} + \frac{1}{k_{eni}}. \end{aligned} \tag{3.2}$$

In the low temperature range, the mean free path of the electron depends on the scattering due to impurities and defects. The impurities and defects depend on the sample and thus mostly temperature independent. The thermal conductivity in this temperature range is proportional to the temperature [37]. The electrical resistivity ρ and k_{eni} are related by the Wiedemann-Franz law where $\rho = \sigma^{-1}$. The equation for k_{eni}, was defined in [56] and simplified as [31]:

$$k_{eni} = \frac{L_o\, T}{\rho}, \tag{3.3}$$

where L_o is the Lorentz constant, ρ is the electrical resistivity of the metal. The "purity level" of a metal is often stated as the Residual Resistivity Ratio (RRR). The larger the RRR value, the fewer impurities and defects in the metal. The RRR is the ratio of the electrical resistivity at room temperature to the electrical resistivity at the boiling point of liquid helium [37, 57]. The RRR is stated as:

$$RRR = \frac{\rho_{295}}{\rho_{4.2}}. \tag{3.4}$$

As the temperature increases, the number of contributing phonons increases. The electron thermal conductivity increases until it reaches a maximum peak, where the electron-phonon scattering becomes dominant. The increase in phonons decreases the electron mean free path due to the increased number of collisions. This results in the electrical thermal conductivity decreasing as the temperature increases. The maximum point depends on the purity of the metal. The simplified thermal conductivity due to scattering of lattice vibrations is stated as [31]:

$$\begin{aligned} \frac{1}{k_{eng}} &= \frac{95.3\, {N_a}^{\frac{2}{3}}\, T^2}{k_{295}\, {\Theta_D}^2} \\ &= a\, T^2, \end{aligned} \tag{3.5}$$

where N_a is the number of effective conduction electrons per atom, Θ_D is

the Debye temperature and k_{295} is the thermal conductivity of the metal at room temperature. Combining (3.2), (3.3), (3.4) and (3.5), the total electron thermal conductivity of the metal in the non-superconducting state is derived as:

$$k_{en} = \left[\frac{\rho_{295}}{L_o\ RRR\ T} + a\,T^2\right]^{-1}. \tag{3.6}$$

3.1.2 Superconductor Region

For $T < T_c$, the electron thermal conductivity enters the superconducting state, k_{es}. In the superconducting region, electrons condense together forming Cooper pairs [58]. Cooper pairs do not contribute to the electrical thermal conductivity of the system and are more resistant to the surrounding vibrations. The electron thermal conductivity decreases with an increase in Cooper pairs. The increase of Cooper pairs below T_c decreases the available electrons in the system [58]. In 1959, Bardeen, Rickayzen and Tewordt used the BCS Theory [58] to develop a model for the thermal conductivity of superconductors [21]. The model develops a function that scales down the thermal conductivity at the lower temperature range due to the formation of Cooper pairs. The ratio is defined as $R(y)$ by Koechlin and Bonin [59] and states:

$$\begin{aligned} R(y) &= \frac{k_{es}}{k_{en}}. \\ &= \frac{1}{F(0)}\left[F(-y) + y\ ln(1+e^{-y}) + \frac{y^2}{2(1+e^{-y})}\right], \end{aligned} \tag{3.7}$$

where k_{es} is the electron thermal conductivity in the superconducting state, k_{en} is the electron thermal conductivity in the normal state and $F(-y)$ is the Fermi integral defined as:

$$F(-y) = \int_0^\infty \frac{z\,dz}{1+e^{(z+y)}}. \tag{3.8}$$

The y term is defined as:

$$\begin{aligned} y &= \frac{\epsilon_0}{k_B\,T} \\ &= \frac{\Delta(T)}{k_B\,T}, \end{aligned} \tag{3.9}$$

where ϵ_0 is gap energy and k_B is Boltzmann's constant. Combining (3.6) and (3.7), the total electron thermal conductivity for $T < T_c$ is derived as:

$$k_{es} = R(y)\left[\frac{\rho_{295K}}{L_o\ RRR\ T} + aT^2\right]^{-1}. \tag{3.10}$$

3.2 Phonon Thermal Conductivity

Phonon thermal conductivity is well known for dielectric solids, where all heat is transported by phonons [37,60,61]. In the normal region for metals, phonon contributions are negligible compared to electron thermal conductivity. When the temperature reduces below the T_c, the phonon conductivity starts to become more dominant. This is due to the increasing number of electrons condensing into Cooper pairs, thus reducing the number of electrons available to conduct heat. The phonon contribution becomes the dominant factor when $T < 0.3\,T_c$ [21].

The phonon conductivity is restrained by the scattering processes contributing to the phonon thermal resistance. Hulm [31] and Markinson [55] lists the four scattering sources for phonons contributing to the total phonon thermal resistance:

- Scattering by conduction electrons.
- Scattering of crystal and grain boundaries.
- Scattering of crystal defects and impurities.
- Scattering of phonons (umklapp process).

For superconducting metals at $T < T_c$, the phonon-phonon scattering process is negligible compared to the other scattering processes [31]. The contribution of the remaining three factors to the total phonon thermal resistance is given as:

$$\begin{aligned} W_{gs} &= \frac{1}{k_{gs}} \\ &= \frac{1}{k_{ge}} + \frac{1}{k_{gcb}} + \frac{1}{k_{gi}}, \end{aligned} \tag{3.11}$$

where k_{gs} it the total phonon conductivity in the superconducting range, k_{ge} is the thermal conductivity contribution from phonon-electron scattering, k_{gcb} is the scattering due to crystal and grain boundaries, k_{gi} is the contribution due to impurities and defects. The thermal resistance for the three scattering processes is computed in the following sections.

3.2.1 Phonon-Electron Scattering

The phonon-electron thermal conductivity has a T^2 dependence in the normal region [55]. In the superconducting region, the phonon-electron thermal conductivity increases as the temperature decreases. The reduction in normal electron density is expressed as $exp(-y)$ where y is defined in (3.9) The

thermal conductivity of phonon conductivity in the superconductivity region was derived by Bardeen, Rickayzen and Tewordt [21]. The phonon conductivity calculation is simplified in [59] as:

$$\begin{aligned} k_{ge} &= exp(y)\frac{27I_3(\infty)\, k_{295K}\, T^2}{\pi^2\, {N_a}^2\, \Theta_D} \\ &= exp(y)\, D\, T^2, \end{aligned} \tag{3.12}$$

where D is the phonon scattering due to electrons, N_a is the number of conduction electrons per atom and $I_3(\infty)$ is the third order Grüneisen integral. The third order Grüneisen integral was calculated as $I_3(\infty) = 7.2$ by [21].

3.2.2 Phonon-Defect Scattering

The phonon-defect scattering region includes scattering due to the crystal and grain boundaries, defects, and impurity. At temperatures lower than the Debye temperature of the metal, the phonon mean free path is larger than the defects and boundaries of the crystal structure. The mean free path is thus independent to the change in temperature [37]. The phonon-defect thermal conductivity is proportional to the heat capacity of the metal and stated as:

$$k_{gd} \propto C_g \propto T^3, \tag{3.13}$$

where k_{gd} is the phonon-defect thermal conductivity given as:

$$k_{gd} = k_{gcb} + k_{gi}. \tag{3.14}$$

The thermal conductivity due the crystalline boundaries was calculated by Casimir [62] and simplified into a general form valid for defects, impurities and crystal boundaries as [59]:

$$\begin{aligned} k_{gd} &= \frac{3.6 \times 10^9}{(a_0\, \Theta_D)^2} \\ &= BlT^3, \end{aligned} \tag{3.15}$$

where l is the phonon mean free path and, a_0 is the lattice parameter of the metal B is the scattering due to crystal boundaries. The phonon mean free path is limited by defects, impurities and crystal boundaries scattering. Combining (3.11), (3.12) and (3.15), the phonon thermal conductivity for superconducting metals are calculated as:

$$k_{gs} = \left[\frac{1}{exp(y)DT^2} + \frac{1}{BlT^3}\right]^{-1}. \tag{3.16}$$

3.3 Total Thermal Conductivity

The total thermal conductivity for superconducting metals is derived by substituting the electron component (3.6) and the phonon component (3.10) into the total thermal conductivity equation (3.1). The total thermal conductivity equation is valid for $T < T_c$.

$$k_s = R(y)\left[\frac{p_{295K}}{L\ RRR\ T} + aT^2\right]^{-1} + \left[\frac{1}{exp(y)DT^2} + \frac{1}{BlT^3}\right]^{-1}. \tag{3.17}$$

The contribution of the electron and phonon part of the total thermal conductivity depends on the temperature of the material. The dominant scattering factor in each temperature region influences the total thermal conductivity of that region. For temperatures close to T_c, the main contributing factor is due to the scattering of normal electrons from impurities and lattice defects [37]. Phonon scattering is insignificant compared to scattering of normal electrons in this temperature region.

For $0.3\,T_c < T < T_c$, the scattering of normal electrons decreases as the contributions due to phonon scattering becomes more dominant with a decrease in temperature. The normal electron scattering is still dominant but its influence decreases with temperature. The decrease in scattering of normal electrons is due to the formation of Cooper pairs.

For $0.2\,T_c < T < 0.3\,T_c$, the phonon scattering becomes the dominant factor influencing k. The phonon mean free path decreases with temperature resulting in a decrease in the number of phonons with temperature. The total thermal conductivity decreases with an increase in temperature in this region [21].

For $T << 0.2\,T_c$, the phonons are primarily scattered by crystal boundaries and defects due to the decrease in the number of phonons. In this temperature range, $k_s \propto T^3$ [33, 62].

3.4 Thermal Conductivity of Material Layers

The thermal conductivity depends on the material properties and the temperature of the material. The thermal conductivity is calculated for some of the materials used in superconductor circuits [6]. The larger the thermal conductivity value, the better the heat conduction. The following section calculates the thermal conductivity of Nb, SiO_2, Mo and Si.

3.4.1 Niobium

Niobium is the superconductive material used in the IC fabrication process for most superconductor components. The popularity of niobium in superconduc-

tor circuits is due to its high T_c compared to any other pure metal (T_c= 9.25K). Since niobium is a Type II superconductor, thermal conductivity is influenced by both magnetic field magnitude and direction. The thermal conductivity of niobium has been widely researched for the pure superconductive, intermediate and normal phase [21, 29, 30, 59, 63–65]. The pure superconductive phase with zero magnetic fields was selected for the thermal conductivity calculations in this study. The condition was selected to focus on heat transfer between layers in the circuit itself.

The thermal conductivity of niobium can be estimated using the theoretically based model for thermal conductivity that was derived in (3.17). The theoretical model relation decouples the electron and phonon components and is valid for $T < T_c$.

The study focuses on the thermal conductivity of niobium in the liquid helium temperature range, $4\,\mathrm{K} < T < T_c$. In this temperature range, the electron scattering contribution is dominant near T_c and decreases as the temperature decreases. With the decrease in temperature, the phonon contribution due to defects becomes more dominant. The phonon bump present at the low temperature range $T < 0.3\,T_c$ is outside the temperature range of interest.

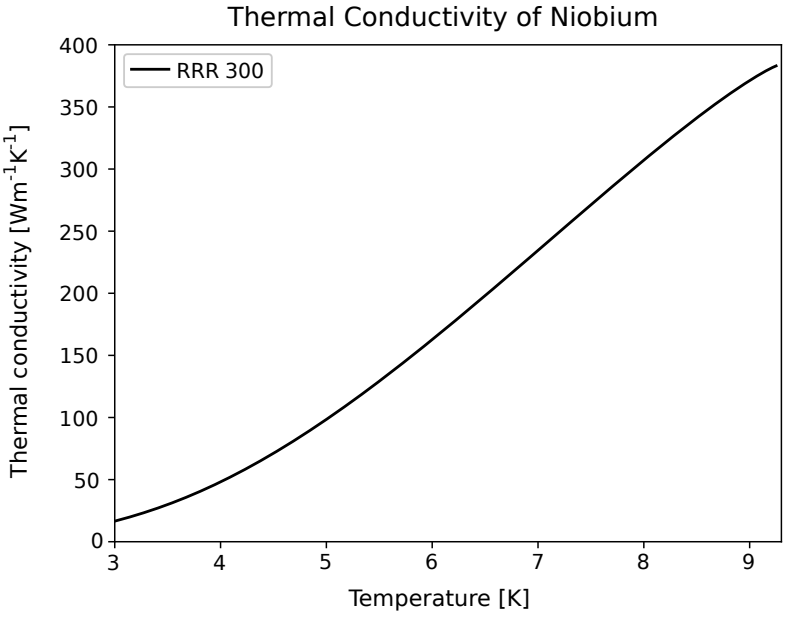


Figure 3.1: The simulated thermal conductivity of niobium for the temperature range 3K-9.3K.

The thermal analysis software calculates the thermal conductivity of niobium using (3.17), RRR = 300, and the best-fit constant values for L, a, D and B [59]. Figure 3.1 shows the simulated thermal conductivity of niobium for the temperature range 3K-9.3K. The best-fit values change depending on

the grade and purity of niobium used. The process history of the material and amount of impurities in the niobium influence the thermal conductivity [57]. The process history includes any heat treatment, deformation, etc [66,67]. The RRR variable is an indication of the purity level of niobium samples used in the simulation. The larger the RRR value, the fewer impurities and defects in the niobium sample. The increase of the thermal conductivity is due to the decrease in scattering from defects. Figure 3.2 shows the influence of RRR on the thermal conductivity of niobium for the temperature range 3K-9.3K.

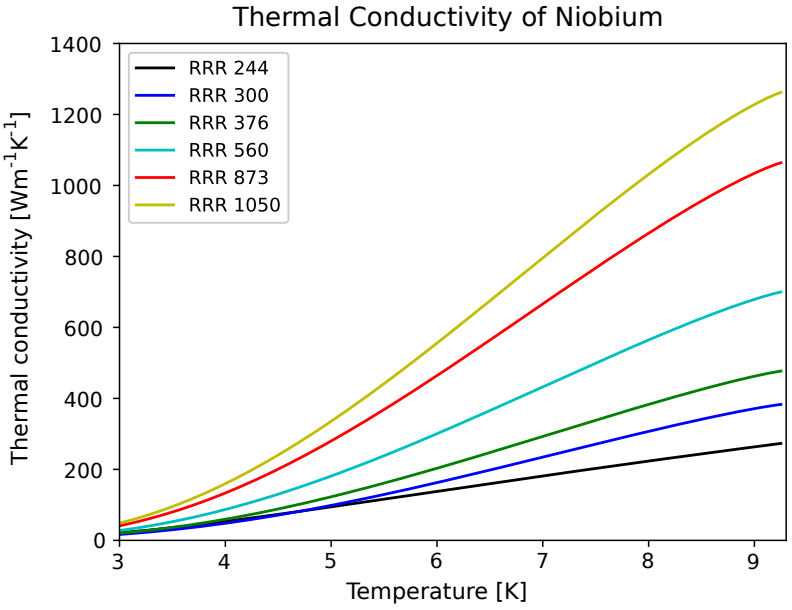


Figure 3.2: The influence of various RRR values on the thermal conductivity of niobium. The thermal conductivity vs. temperature for the temperature range 3K-9.3K.

3.4.2 Silicon Dioxide

According to the SFQ5ee process, the SiO_2 dielectric is deposited using the PECVD method [6]. The thermal conductivity of SiO_2 is assumed to be a constant value equal to $k = 1.1$ (W m^{-1} K^{-1}) [68,69]. The thermal conductivity of the SiO_2 depends on the thickness and temperature of the sample. The SiO_2 layer thicknesses differ depending on the location of the layer on the chip. The difference in the temperature, gas pressure, flow rate and RF power could influence the thermal conductivity properties. More data is required for measurement in the low-temperature range $T < 10$K.

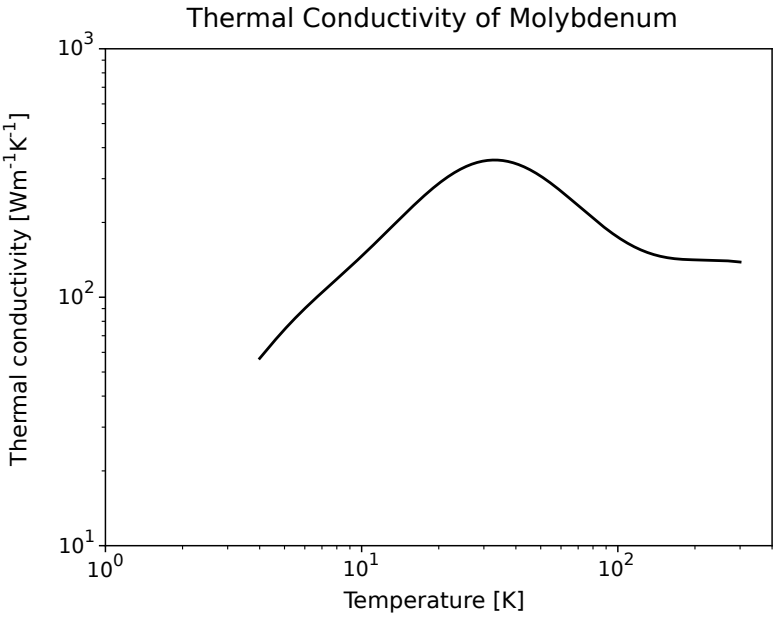


Figure 3.3: Thermal conductivity of Molybdenum for the temperature range 3-300K using [1].

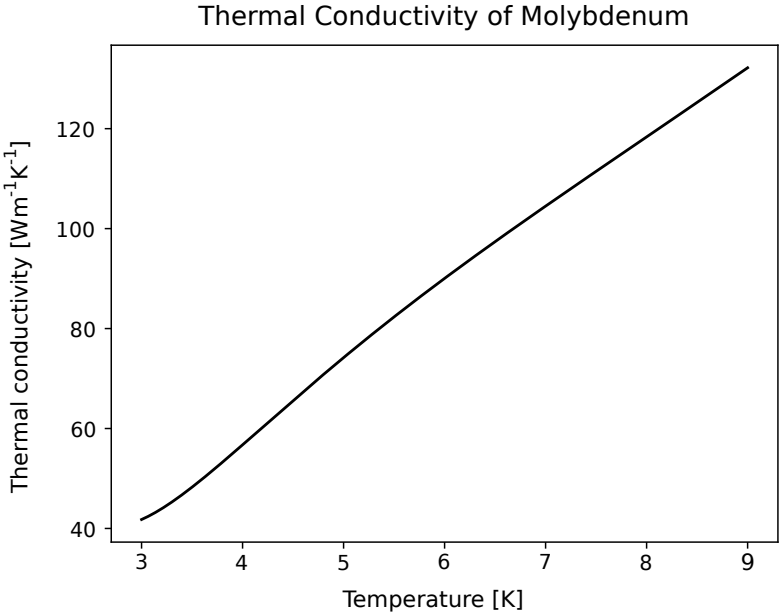


Figure 3.4: Thermal conductivity of Molybdenum for the temperature range 3-9K using [1].

3.4.3 Molybdenum

The non-superconducting Mo layer is the resistive layer utilised in the SFQ5ee design process, and is also called the standard sheet resistance (SSR) layer [6]. The resistive layers in the RSFQ circuit are used for shunting JJs, biasing the

DC current and RF impedance matching [13]. The heat dissipated by the Mo resistor increases the temperature of the surrounding material and the Nb vias in contact with the resistive layer. The increase in current increases the heat dispersed by the Mo resistive layer. The thermal conductivity of Mo influences the heat propagation through the resistor layer and the transfer of heat to the contacting niobium vias.

The thermal conductivity of Mo changes with temperature and is derived from measured sample data as per article [70, 71]. The curve fit equation describing the change in thermal conductivity with temperature was derived by NIST [1]. The curve fit equation is valid over the temperature range 3-300K. Figure 3.3 shows the thermal conductivity of Mo over the temperature range 3-300K. Figure 3.4 shows the simulated thermal conductivity of molybdenum for the temperature range 3K-9K using the curve fit equation.

3.4.4 Silicon

The various phonon scattering mechanisms mainly affect the thermal conductivity of Si. In the low temperature range, the scattering due to the crystal boundaries governs the thermal conductivity. The thermal conductivity increases proportionally to the specific heat of Si until the maximum peak is reached. Past the maximum point, the thermal conductivity decreases due to the Umklapp scattering. The maximum point is limited by the number of impurities and imperfections present in the material [37].

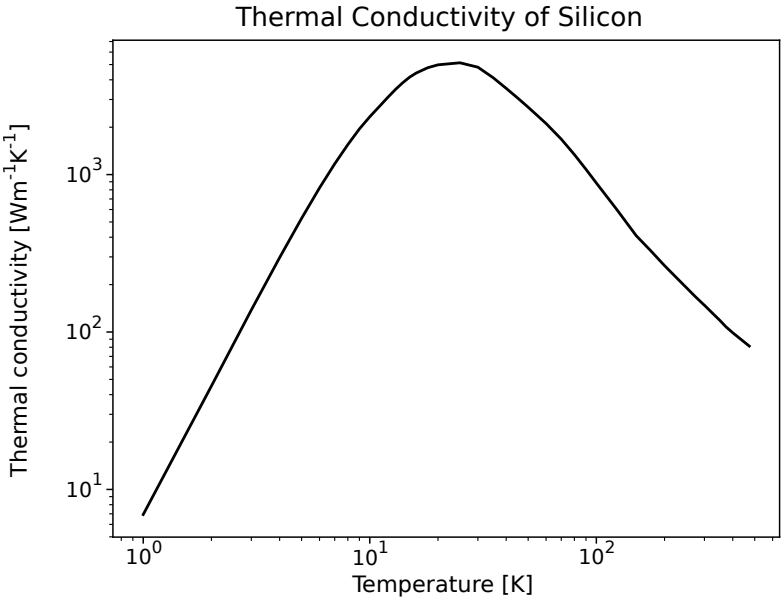


Figure 3.5: The thermal conductivity of pure Silicon for the temperature range 2-300K.

For the SFQ*ee process, the superconductor IC is fabricated on a 200mm Si wafer [6]. The superconductor circuit is built up layer for layer on top of the Si wafer.

The thermal conductivity of pure silicon was measured by multiple authors [72, 73]. The curve fit equation derives the thermal conductivity with a change in temperature from the recommended values for Si [74]. The curve fit equation is incorporated into the thermal analysis tool and used over the temperature range 2-300K. Figure 3.5 shows the variation of thermal conductivity with temperature plot of highly pure silicon for 2-300K.

3.5 Summary

The chapter was dedicated to deriving the thermal conductivity at low temperatures for superconducting circuitry. The total thermal conductivity for superconducting metals was derived from theory. The contribution as a result of the electron and phonon conductivity was discussed for both the superconducting and non-superconducting region. The electron contribution dominates for the higher temperatures near T_c. As the temperature decreases, the electron contribution decreases due to the formation of Cooper pairs until the phonon contribution dominates. It was found that the electron scattering contribution was dominant for the temperature region under investigation, $4K < T < 9.2K$.

The total thermal conductivity equation was used to calculate the thermal conductivity of Nb. The thermal conductivity of Mo, Si and SiO_2 was calculated from theory. Several factors including purity, flaws, magnetic field, manufacturing process, etc. limit the thermal conductivity model. With all the limitations as discussed in the chapter, it is difficult to derive a thermal conductivity model for all cases. The data indicated that a slight increase in temperature has a major impact on thermal conductivity for $T < T_c$. The larger thermal conductivity allows more heat to transfer through the circuit.

Chapter 4

Numerical Thermal Analysis Model

4.1 Introduction

This chapter derives the variational formulation, heat source and boundary conditions to solve the heat conduction PDE (Partial Differential Equation) iteratively using FEM. The heat conduction PDE is derived using the temperature-dependent thermal conductivity derived in Chapter 3. The variational formulation is calculated from the nonlinear heat transfer PDE and solved iteratively using Newton's method. The heat source is the heat generated in the bias resistor due to Joule heating and is further explained later in this chapter. The general formula for each boundary condition is derived and added to the variational formulation. The applied boundary conditions depend on the thermal load applied to the circuit. FEniCS [49] is selected for the thermal analysis, because of the easy to use a Python interface, support for multiple finite element solvers and powerful parallel processing capabilities.

4.2 Heat Conduction Equation

The heat conduction equation calculates the propagation of heat flux through a medium over time [34]. By substituting the total thermal conductivity into the heat conduction PDE, the temperature can be calculated at any point in the medium. The heat conduction PDE is derived from Fourier's law of heat transfer and the first law of thermodynamics. Fourier's law states that the heat transfer through an object is proportional to the product of the negative temperature gradient and the area perpendicular to the heat flow [24, 34]. Fourier's law of thermal conduction is defined as:

$$\dot{q} = \frac{\dot{Q}}{A} = -k(T)\nabla T, \tag{4.1}$$

where $\dot{Q}$ is the rate of heat transfer, A is the area normal to the heat transfer, $k(T)$ is the temperature-dependent thermal conductivity of the material and ∇T is the temperature gradient. According to the first law of thermodynamics, the change in the system's total energy depends on the sum of heat entering the system and work done on the system [24]. The general heat conduction equation is derived by combining (4.1) and the first law of thermodynamics.

$$pc_p \frac{dT}{dt} = \nabla \cdot (k(T)\nabla T) + q, \tag{4.2}$$

where p is the density, c_p is the specific heat capacity of the material and q is rate of heat generation per unit volume . The general heat equation is also known as the Fourier-Boit equation, reducing to the Poisson equation for the steady state case. For the heat conduction simulation, we make use of the steady state case of (4.2) stated as:

$$\nabla \cdot (k(T)\nabla T) = -q. \tag{4.3}$$

The steady state case assumes the temperature in the system is unaffected by time ($dT/dt = 0$) [24, 34]. With the thermal conductivity dependent on the temperature T, the PDE generalising the heat equation is nonlinear. The steady-state equation provides the means to investigate the effect of the temperature-dependent thermal conductivity on the structure.

4.3 PDE Model Problem

The model for the non-linear steady-state heat conduction problem with multiple Dirichlet, Neumann and Robin boundary conditions state:

$$-\nabla \cdot (k(u)\nabla u) = f \quad \text{on} \quad \Omega, \tag{4.4}$$

$$u = u_D^i \quad \text{on} \quad \Gamma_D^i, \tag{4.5}$$

$$-k\frac{\partial u}{\partial n} = g^i \quad \text{on} \quad \Gamma_N^i, \tag{4.6}$$

$$-k\frac{\partial u}{\partial n} = h^i(u - T_s^i) \quad \text{on} \quad \Gamma_R^i. \tag{4.7}$$

where i = 1,2,... is the number of the boundaries where each boundary condition is applied. Γ_R^i, Γ_N^i and Γ_D^i are the parts of the ith number of boundary where the Robin, Neumann and Dirachlet boundaries are applied respectively. $\Gamma_R^i \cup \Gamma_N^i \cup \Gamma_D^i$ form part of the boundary $\partial\Omega$. The different types of boundaries and the application of each is discussed later in Section 4.9.

4.4 Variational Formulation

The steady-state form of the heat conduction PDE was calculated previously as per (4.3). The heat conduction PDE with boundary and initial condition is known as the "strong form" of the problem. The finite element method cannot use the PDE in its present form and must be restated in its integral form known as variational or weak formulation. The strong and weak form of the problem is equivalent [46, 75]. The variational formulation of the heat conduction PDE is solved using FEM. The Galerkin method for differential equations is utilised to calculate the variational form. The variational formulation for the 1D and 3D case of the heat conduction PDE are determined in the following sections. The unknown function, T, is replaced with the trial function, u. This is valid over the boundary Ω.

4.4.1 1D Variational formulation

The variational formulation across a plane wall with a finite length in the x-direction and infinite length in the y-direction is calculated. The steady-state heat conduction PDE from (4.3) is reformatted into a 1D equation:

$$-\frac{d}{dx}\left(k(u)\frac{du}{dx}\right) = f(x) \quad x \in \Omega = [0, L], \tag{4.8}$$

where $k(u)$ is the temperature dependent thermal conductivity, u is the trial function for the approximation and is valid over the domain $x \in \Omega = [0, L]$. The equation is multiplied by the test function, v, and integrated over the domain:

$$-\int_0^L \frac{d}{dx}\left(k(u)\frac{du}{dx}\right)v dx = \int_0^L f v dx. \tag{4.9}$$

The equation is integrated by part to reduce derivatives from second order to first order, where the test function, v, and the trial function, u, must be valid over the test function. Find $u \in V$ such that:

$$\int_0^L k(u)\frac{du}{dx}\frac{dv}{dx}dx = \int_0^L f v dx + \left[k(u)v\frac{du}{dx}\right]_0^L \quad \forall v \in V_0, \tag{4.10}$$

where the trial, V, and test, V_0, function space are defined as:

$$V = \{v \in H^1(\Omega)\}, \tag{4.11}$$

$$V_0 = \{v \in H^1(\Omega); v = 0 \quad \text{on} \quad \Gamma_D\}. \tag{4.12}$$

where $H^1(\Omega)$ is the Sobelev space used for the test and trial function. The final term in (4.10) is the boundary function applied to the plane wall. The boundary points of interest for the plane wall are located at $x[0]$ and $x[L]$. The first order derivative with the applicable boundary conditions is solved iteratively using Newton's Method.

4.4.2 3D Variational Formulation

The one-dimensional case is sufficient for simple problems where the heat transfer is negligible in the other dimensions. For heat transfer problems with more complex structure and multiple boundary conditions, the heat transfer in three-dimensions is more important. The nonlinear Poisson equation for the three-dimensional steady-state heat conduction problem states:

$$-\nabla \cdot (k(u)\nabla u) = f, \quad x \in \Omega \subset \mathbb{R}^3, \tag{4.13}$$

where:

$$\nabla \cdot (k(u)\nabla u) = \frac{\partial}{\partial x}\left(k(u)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k(u)\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(k(u)\frac{\partial u}{\partial z}\right). \tag{4.14}$$

The test function, v is multiplied by (4.13) and integrated over the domain of the PDE:

$$-\int_\Omega \nabla \cdot (k(u)\nabla u)v dx = \int_\Omega f v dx, \tag{4.15}$$

where the test function, v, and the trial function, u, is valid over the test function such that $u \in V$. The variational formulation calculation between the two dimensions differs due to the integration by parts used for the calculation. The equation is integrated by part using Green's first identity. The identity is stated as:

$$-\int_\Omega \nabla \cdot (k(u)\nabla u)v dx = \int_\Omega k(u)\nabla u \cdot \nabla v dx + \int_{\partial\Omega} k(u)\frac{\partial u}{\partial n} v ds, \tag{4.16}$$

where $\partial\Omega$ is the boundary of Ω, $\frac{\partial u}{\partial n} = \vec{n} \cdot \nabla u$ is the derivative of the trial function normal to the surface. The resulting variational formulation of the heat conduction equation is calculated using Galerkin's method:

$$\int_\Omega k(u)\nabla u \cdot \nabla v dx = \int_\Omega f v dx + \int_{\partial\Omega} k(u)\frac{\partial u}{\partial n} v ds \quad \forall v \in V_0. \tag{4.17}$$

where the trial, V, and test, V_0, function space are defined as:

$$V = \{v \in H^1(\Omega)\}, \tag{4.18}$$
$$V_0 = \{v \in H^1(\Omega); v = 0 \quad \text{on} \quad \Gamma_D\}. \tag{4.19}$$

where $H^1(\Omega)$ is the Sobelev space used for the test and trial function. The integral $\int_{\partial\Omega}()ds$ of (4.17) is the boundary term where the boundaries conditions are applied to the function. The variational formulation is implemented into FEniCS together with the boundary conditions, initial conditions and source term. The heat transfer problem is solved iteratively using Newton's Method.

4.5 Newton's Method

Iterative methods can be utilised to solve the variational form of the nonlinear PDE. The nonlinear algebraic equation is solved using Newton's method. Newton's method, also known as the Newton-Raphson method, approximates the roots of a function using an iterative process [47]. The variation form from (4.17) can be restated in the following general and compact form:

$$F(u; v) = 0 \quad \forall v \in V. \tag{4.20}$$

The nonlinear equation can be solved by using Newton's method to linearise the equation. This is done by approximating $F(u)$ by its Taylor series expansion around a known guess value $\tilde{u}$ and truncating the nonlinear part [76].

$$F(u) \approx F(\tilde{u}) + J(\tilde{u}) \cdot (u - \tilde{u}) = \hat{F}(u), \tag{4.21}$$

where $\hat{F}(u)$ is the linear equation and J is the Jacobian matrix of F defined as:

$$J_{i,j} = \frac{\partial F_i}{\partial u_j}. \tag{4.22}$$

The linear equation is thus solved with respect to the vector δu for $\delta u \in \mathbb{R}^n$

$$J\delta u = -F(\tilde{u}). \tag{4.23}$$

The improved guess value depending on δu is calculated as:

$$u = \tilde{u} + \omega\delta u. \tag{4.24}$$

The process iterates until the improved guessed value converges to the actual solution.

4.6 Error Estimate

The true value for the heat conduction through superconductor circuits are often unknown. The process to design and simulate the superconductor circuit are both costly and time-consuming. Since the true value is often unknown, the approximate value quantifies the errors in the simulation. Solving the heat conduction PDE iteratively using FEM, gives an approximate solution at the end of the iteration [77]. During the iteration process, the current and previous approximated solution are stored. The approximation error is the difference between the two approximated solutions:

$$E_a = u - u_p, \tag{4.25}$$

where E_a is the approximation error, u is the current approximated solution and u_p is the approximated solution from the previous iteration. The relative approximation error of the FEM solution is calculated by dividing the approximate error from (4.25), by the approximate solution to the current iteration:

$$e_a = \frac{E_a}{u}. \tag{4.26}$$

The relative approximation error decreases as the solution converges to a better approximation. The approximation improves with each iteration up to a desired tolerance for the relative approximation error. The tolerance is usually a percentage of the absolute relative approximation error and calculated as:

$$|e_a| \leq \text{tolerance}. \tag{4.27}$$

where the tolerance is 0.001 unless otherwise specified for the thermal simulations. The simulation iterates until the desired approximation error is reached.

4.7 Heat Generation

The source term, f, from (4.17) represents the rate of energy generation per unit volume in the domain. The thermal energy generation due to Joule heating in the resistive layers of the circuit is of interest. The heat generated by the DC bias current through the bias Mo resistor is the internal heat source for the simulation. Joule heating (also known as ohmic or resistive heating) is the generation of thermal energy from the electrical current passing through the

resistive element [24]. The heat energy generated in the resistor is dissipated to its surroundings, increasing the temperature of the material surrounding the resistor. The amount of heat dissipated from the resistive layer per unit time is:

$$Q = I^2 R, \tag{4.28}$$

where I is the electric current through the resistor and R is the resistance of the Mo thin-film resistor. The resistance of the thin-film resistor is calculated using the sheet resistance defined as [3]:

$$R = R_s\left(\frac{L}{w}\right), \tag{4.29}$$

where R_s is the sheet resistance measured in (Ω/square), w is the width and L is the length of the resistor between the vias connecting the resistor to the Nb wire layer. Substituting (4.29) into (4.28):

$$Q = I^2 R_s\left(\frac{L}{w}\right). \tag{4.30}$$

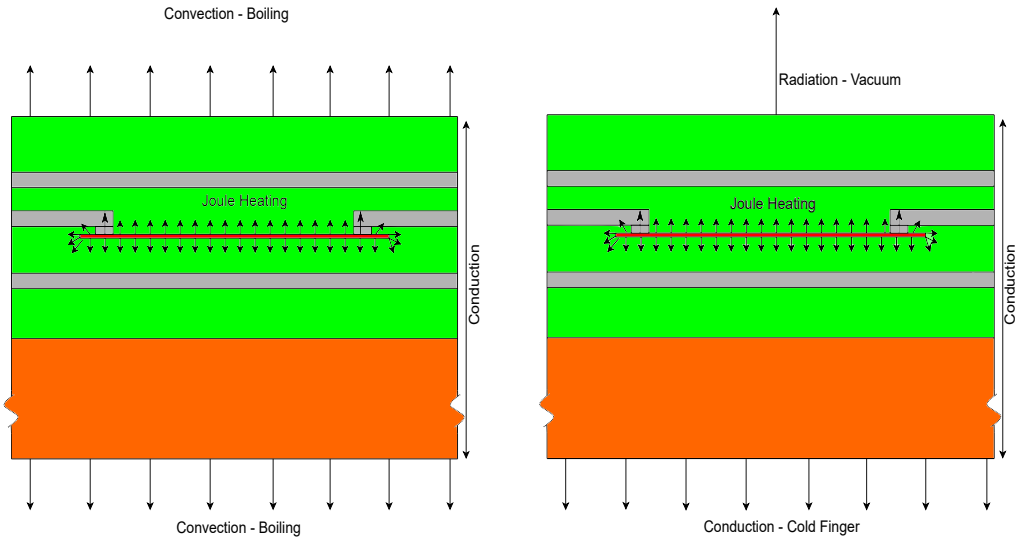
The heat dissipation equation is applied to each bias resistor in the circuit. The current through the resistor is the limiting factor of the bias resistor. If the current through the resistor is too large, the heat generated by the resistor will increase the temperature of the resistor-contact vias past the T_c point. The increase in temperature past the T_c point will result in the niobium layer exiting the superconducting state [3, 4].

4.8 Thermal Load

The thermal load shows the heat transfer processes both within the structure and with the surrounding environment. The thermal load of a bias resistor segment is cooled using the pool bath as reflected in Figure 4.1a. The cold finger method is displayed in Figure 4.1b. The boundary conditions added to the variational formulation depends on the heat transfer from the external environment. Each of the heat transfer mechanisms added to the variational formulation helps solve the heat transfer problem more accurately.

4.8.1 Conduction

Thermal conduction is one of the primary forms of heat transfer through a medium or mediums in direct contact. Figure 4.1 shows the heat conduction from a heated source to the surrounding layers. The heat conduction rate from



(a) Helium Pool Bath Schematic. (b) Cold Finger Schematic.

Figure 4.1: The thermal schematic of a bias resistor segment cooled using the pool bath and Cold Finger method.

(2.11) depends on the thermal conductivity of the material and surface area of the medium at the connection point. The heat dissipation is more pronounced in the high thermal conductivity metal layers compared to the lower thermal conductivity dielectric layers. Heat transfer efficiency depends on surface area size between mediums and thermal contact at the interface. The heat dissipates throughout the medium until the model reaches equilibrium.

The layers of the thermal schematics in Figure 4.1 are fully planarized. The planarization of the layers is achieved by chemical mechanical planarization (CMP) of the interlayer SiO_2 dielectric [3, 6]. The planarized surfaces are assumed to be smooth and make perfect thermal contact at the interface.

Figure 4.2 shows the interface boundary condition for two mediums in perfect thermal contact. The boundary conditions for conduction are modelled as a constant temperature or heat flux applied to the boundary. The temperature at the contact point is identical for both materials and the interface between the layers may not store any energy. Thus, the same heat flux propagates from the one side to the other. The heat flux through the interface between the layers is derived from Fourier's law [24].

$$\dot{Q}_1 = \dot{Q}_2, \tag{4.31}$$

with

$$-k_1 \frac{\partial T_1(x_0, t)}{\partial x} = -k_2 \frac{\partial T_2(x_0, t)}{\partial x}, \tag{4.32}$$

where k_1 and k_2 is the thermal conductivity material 1 and 2, respectively.

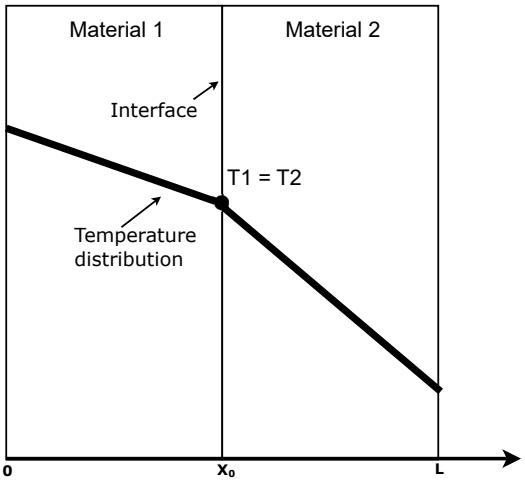


Figure 4.2: Interface boundary condition and temperature distribution for two mediums in perfect thermal contact.

This equation is for the ideal thermal contact case and does not result in a temperature drop at the interface. The thermal contact resistance is outside the scope of this thesis.

4.8.2 Convection

Convection is the transfer of energy from a solid medium to an adjacent liquid or gas due to the motion of molecules in the fluid. Convection is a combination of the effects due to conduction and fluid motion [24]. Figure 4.1a shows the heat transfer through a circuit cooled by helium bath submersion. The superconductor circuit is cooled using the pool boiling heat transfer model. Using pool boiling, the superconductor circuit is submerged in a bath of helium kept at a constant temperature [78,79]. Helium is the cryogenic fluid utilised to cool superconductor electrons to cryogenic temperatures due to its low boiling point of 4.2K. The He bath method of cooling is a common method used due to its simplicity and effectivity [80].

The heat transfer to the surrounding helium bath is dependent on multiple factors including temperature, pressure, frequency of the heat flux, orientation, and roughness of the surface in contact with the bath [79–81]. The simulation assumes the temperature of the helium pool is at the saturation temperature ($T_{sat} = 4.2$) and one atmosphere of pressure. Pressure points other than the 1 atm affect the reliability of the sample data [79]. The heat transfer of the helium in contact with the surface is explored in more detail in the following section.

4.8.2.1 Boiling Heat Transfer

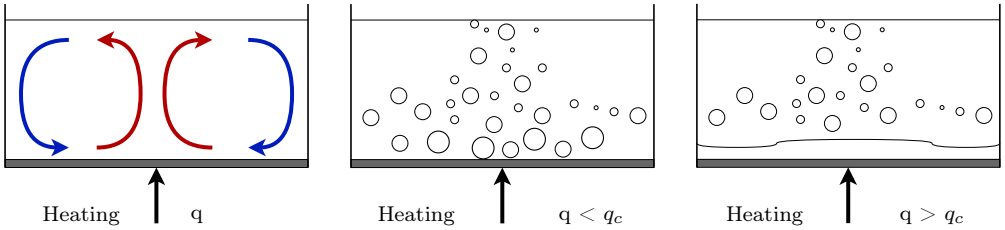
Boiling is the liquid to vapour phase change, as the temperature increases past the saturation point of the liquid [24]. The saturation point is the temperature where the liquid starts to boil and depends on the liquid properties. The saturation point of the He 1 used in the He pool bath, is 4.2K. Boiling convection takes place at the solid-liquid boundary of a heated surface where the temperature of the surface is greater than the saturated temperature of the liquid [80]. Boiling heat transfer depends on the enthalpy of vaporisation of the liquid and the surface tension between the solid-liquid interface. During the boiling process, vapour bubbles form on the solid-liquid interface. The vapour bubbles detach from the surface and propagate upwards depending on the pressure in the vapour bubble [24, 80]. The heat transfer due to boiling is a highly effective form of heat transfer since the cooler surrounding fluid occupies the gap left behind by the rising bubbles. This results in a decrease in the surface temperature in the nucleating region. Boiling convection heat transfer is expressed as the heat flux transferred out of the surface into the liquid using Newton's law of cooling [24] as stated in (2.12):

$$\dot{Q} = hA(T_s - T_\infty). \tag{4.33}$$

The heat transfer coefficient, h, for boiling heat transfer is much larger than that of other forms of convection. The excess heat between the surface and the surrounding liquid is shown in (4.33). Boiling is classified as either pool boiling or forced convection boiling depending on the presence of bulk fluid motion [24]. The heat transfer due to pool boiling is of interest due to the lack of bulk fluid motion when using a He bath.

4.8.2.2 Pool Boiling

Pool boiling heat transfer is a common engineering problem encountered when working with cryogenic fluids such as liquid helium. Pool boiling is boiling heat transfer without any externally added bulk fluid motion. The only fluid flow present in pool boiling is due to natural convection and bubble motion. Pool boiling can be viewed as a heated surface in a large bath of fluid, where the surface is negligible compared to the pool size [24, 81]. The pool boiling method for He I take on three different boiling regions depending on several variables influencing the surface and liquid. These include the pressure, bath temperature, surface properties and orientation to name a few. The three boiling regions are: natural convection, nucleate boiling and film boiling region [78, 81]. Figure 4.3 shows the three boiling regions for He liquid. From the three boiling regions, the nucleating boiling region is of interest.



(a) Natural Convection. (b) Nucleate Boiling. (c) Film Boiling.

Figure 4.3: The different boiling regions in pool boiling.

4.8.2.3 Nucleate Boiling

When the surface temperature increases past the saturation point, bubbles begin to nucleate on the surface and then enter the nucleate boiling region. Using liquid helium, small vapour bubbles begin to form on the surface after a small increase in heat flux (W/m^2) [81]. In the nucleate boiling region, a layer of superheated liquid is formed around the surface. Bubbles begin to form at nucleation sites around the superheated surface and detach to the surrounding liquid. The surrounding cooler fluid fills the space left by the bubble and cools the surface. The turbulence created near the surface due to bubble formation increases surface-to-liquid heat transfer. The heat dissipated by the surface, is quickly and efficiently carried into the surrounding liquid. The nucleate boiling heat transfer depends on the fluid dynamics of bubble growth and detachment [24, 81]. For a slight increase past the saturation point, the bubbles formed lose heat rapidly and are reabsorbed back into the bulk fluid. As the temperature increases further, the rate of bubble formation increases rapidly. The larger bubbles reach the surface where it releases the vapour to the surroundings.

Nucleate boiling heat transfer depends on several properties including surface nucleation area size, bubble formation and detachment rate, number of surface imperfections, etc. These factors complicate the development of a theoretical relationship for the heat transfer process in the nucleate boiling range [24, 81]. The correlation that best describes the nucleate boiling heat transfer was proposed by Kutateladze [82]. The Kutateladze correlation fit the experimental data and the best approximation for the nucleating region [78, 80, 81, 83]. The Kutateladze correlation is valid for many different cryogenic fluids and states:

$$\frac{h}{k_1}\left(\frac{\sigma}{g\rho_l}\right)^{\frac{1}{2}} = 3.25\times10^{-4}\left[\frac{qC_{pl}\rho_l}{h_{lv}\rho_v k_l}\left(\frac{\sigma}{g\rho_l}\right)^{\frac{1}{2}}\right]^{0.6}\left[g\left(\frac{\rho_l}{\mu_l}\right)\left(\frac{\sigma}{g\rho_l}\right)^{\frac{3}{2}}\right]^{0.125}\left(\frac{P}{(\sigma g\rho_l)^{\frac{1}{2}}}\right)^{\frac{3}{2}}, \tag{4.34}$$

where σ is the surface tension, g is the acceleration due to gravity, ρ is the density, P is pressure, C is specific heat, μ is the Newtonian coefficient of

of viscosity and h_{lv} is the latent heat of vaporization. The subscripts l and v represents the 'liquid' and 'vapour' respectively. The complex expression (4.34) can be reorganized into a more manageable form to calculate the q [81].

$$q = 1.90\times10^{-9}\left[g\left(\frac{\rho_l}{\mu_l}\right)^2\chi^3\right]^{0.3125}\left(\frac{P\chi}{\sigma}\right)^{1.75}\left(\frac{\rho_l}{\rho_v}\right)^{1.5}\left(\frac{C_p}{h_{fg}}\right)^{1.5}\left(\frac{k_l}{\chi}\right)(T_s-T_b)^{2.5}, \tag{4.35}$$

where:

$$\chi = \left(\frac{\sigma}{g\rho_l}\right)^{0.5}, \text{and} \tag{4.36}$$

where T_b and T_s are the bath and surface temperature respectively. In the nucleating boiling region, the heat flux can be calculated by utilising the Katateladze correlation. The heat flux derived using Kutateladze [82] is proportional to the power function of the change in temperature:

$$q = \varphi\Delta T^n, \tag{4.37}$$

where φ and n are constants that depend on the pressure, surface properties and orientation [79, 81]. The nucleate boiling boundary condition is added to the veriational formulation using Robin boundaries. The Kutateladse correlation for He I, at 4.2K, and a pressure of 1 atm, on a flat surface facing upward, was calculated as [81]:

$$q = 5.8(T_s - T_{surr})^{2.5} \quad (\mathrm{W/cm^2}). \tag{4.38}$$

With an increase in heat flux, bubble formation increases rapidly and becomes more erratic. The rate of bubble formation increases until a vapour film forms around the surface. The maximum point between nucleating boiling and film boiling is the critical heat flux point. The correlation (4.35) is valid for a heat flux lower than q_c, the maximum point. The critical heat flux derived by Kutateladze [82] and Zuber [84] produced comparable results based on experimental data. The critical heat flux equation is stated as [78, 81]:

$$q_c = 0.16h_{lv}p_v^{1/2}[\sigma g(p_l - p_v)]^{1/4}. \tag{4.39}$$

The predicted critical heat flux calculated using (4.39) for a bath of liquid helium is $q_c = 8.5\ \mathrm{kW/m^2}$ [81]. The predicted value was calculated for an upward surface in a bath of 4.2K liquid helium at a pressure of 1 atm. It is best to avoid the critical heat flux point since the heat transfer rate is greatest in the nucleating boiling range for a slight increase in temperature.

4.8.3 Radiation

Thermal radiation is the energy emitted or absorbed by a surface in the form of electromagnetic waves. The amount of thermal radiation emitted from the surface depends on the electromagnetic wavelength at a specific temperature and the roughness of the surface. Thermal radiation significantly influences cryogenic systems since it is the only heat transfer mechanism that functions in a vacuum [24, 80]. The surface of the circuit from Figure 4.1b radiates the energy from the surface. The surface is modelled to be surrounded by a large black body surface that does not influence the net radiation heat transfer. The radiation heat transfer between two surfaces was stated earlier in (2.13) as:

$$\dot{Q}_{rad} = \sigma \varepsilon A_s ({T_s}^4 - {T_{surr}}^4). \tag{4.40}$$

The surface of the circuit is assumed to have a low emissivity value for the simulation. The value of emissivity is influenced by temperature, impurities, oxidation, and roughness of the surface. The lowest emissivity value can be achieved by having a polished, clean surface free from impurities [80]. The constant value for emissivity is used since the change in temperature is exceedingly small for the simulation. The thermal radiation is added to the FEM model making use of Robin boundaries.

$$E = \sigma \varepsilon T^4. \tag{4.41}$$

The surface of the circuit under test is assumed to be a smooth and polished surface with $\varepsilon = 0.018$.

4.9 Boundary Conditions

The external conditions acting upon the boundaries of the domain is known as boundary conditions. The boundary conditions constrain the model and must be applied to solve the PDE using FEM. Figure 4.1 shows the convective, conductive and radiation boundaries that effect the circuit under test.

Boundary conditions influence the results of the finite element analysis (FEA). Badly implemented boundary conditions can cause the FEA to converge to an incorrect solution or even ultimately diverge [49]. The following section describes the boundary types and boundary application function.

4.9.1 Boundary Types

The heat transfer problem depends on the applied boundary conditions to find a solution. The test conditions of the circuit influence the applicable boundary conditions. Figure 4.3 shows the different heat transfer mechanisms contributing to the heat transfer in the superconductor circuit. The circuit is

influenced by the conductive, convective and radiation heat transfer applied to the boundaries of the model. Figure 4.4 shows an example of the domain to which the boundary conditions are applied. The applicable boundary conditions are restated in the general Dirichlet, Neumann and Robin boundary conditions form.

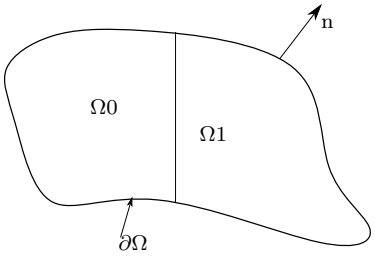


Figure 4.4: The domain and boundary example for the heat transfer problem. The different subdomains $\partial\Omega 0$ and $\partial\Omega 1$ represents the layers with different material properties.

4.9.1.1 Dirichlet Boundary

The Dirichlet boundary applies a constant temperature value to the variational form, when the trial function, u, is known on a part of the boundary. The test function, v, disappears on that boundary. This results in $v = 0$ on the boundary when a temperature is specified on that boundary. The boundary condition applied to the system functions as steady heat conduction through the boundary. The general Dirichlet boundary states:

$$u = u_D \quad \text{on} \quad \partial\Omega_D, \tag{4.42}$$

where u_D is the prescribed constant value on the applicable boundary $\partial\Omega_D$ forming part of the boundary of the domain Ω.

4.9.1.2 Neumann Boundary

The Neumann boundary is the derivative of the PDE normal to the boundary of the domain. The Neumann boundary corresponds to the heat flux added to the boundary of the domain for the heat conduction problem. The general Neumann boundary states:

$$-k(u)\frac{\partial u}{\partial n} = g \quad \text{on} \quad \partial\Omega_N, \tag{4.43}$$

where $\vec{n}$ is normal to the boundary, g is the prescribed function over the domain and $\partial\Omega_N$ is part of the boundary on the domain Ω. The Neumann boundary is also called the natural boundary since it naturally appears during

the variational formulation process. The natural boundary is defined as:

$$\frac{\partial u}{\partial n} = \nabla u \cdot n = 0. \tag{4.44}$$

The natural boundary applies when the boundary conditions are omitted from the variational formulation. The natural boundary also applies when only Dirichlet boundaries are applied.

4.9.1.3 Robin Boundary

In Robin boundary models, the heat transfer between the circuit boundaries and the surrounding environment is described. The Robin boundary condition is a combination of the Dirichlet and Neumann boundary and can be used to approximate them. The Robin boundary is used to model the heat transfer due to convection and radiation. The Robin boundary arises naturally from Newton's cooling law and states:

$$-k(u)\frac{\partial u}{\partial n} = h(u)(u - T_s) \quad \text{on} \quad \partial\Omega_R, \tag{4.45}$$

where $h(u)$ is the heat transfer coefficient between the surface and surroundings, T_s is the temperature of the surroundings and $\partial\Omega_R$ is part of the boundary on the domain Ω. $h(u)$ is a nonlinear term and contributes to the Jacobian matrix when using Newton's method.

4.9.2 Boundary Condition Calculation

The last part of the equation from (4.10) and (4.17) is the boundary condition for the one- and three-dimensional variational form respectively. The boundary term function expands as the number of boundary conditions applied increases. The function enables the boundary parts to be added together where, b_i is the number of boundaries on the domain. The sum of the boundary conditions gets added to the boundary term of the variational formulation. The sum of the boundaries makes use of the general form of the Dirichlet, Neumann and Robin boundary. The equations are calculated for $v = 0$ and $i = 1, 2, ..., b_i$:

$$-\int_{\partial\Omega} k(u)\frac{\partial u}{\partial n}vds = \sum_i \int_{\partial\Omega_N^i} k(u)\frac{\partial u}{\partial n}vds + \sum_i \int_{\partial\Omega_R^i} k(u)\frac{\partial u}{\partial n}vds \tag{4.46}$$

Substituting 4.43 and 4.45 into 4.46:

$$-\int_{\partial\Omega} k(u)\frac{\partial u}{\partial n}vds = \sum_i \int_{\partial\Omega_N^i} g_i vds + \sum_i \int_{\partial\Omega_R^i} h_i(u)(u - T_{s_i})vds \tag{4.47}$$

where $\partial\Omega_D, \partial\Omega_N, \partial\Omega_R$ are defined as the Dirichlet, Neumann and Robin boundary parts, respectively. $\partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_R$ form the complete boundary of the variational formulation $\partial\Omega$. The trial function vanishes on the parts where $\partial\Omega_D$ applies. This is due to the fixed values applied to the boundaries. We obtain the general variational formulation with boundary conditions by combining 4.47 with 4.17:

$$\int_\Omega k(u)\nabla u \cdot \nabla v dx = \int_\Omega f v dx + \sum_i \int_{\partial\Omega_N^i} g_i v ds + \sum_i \int_{\partial\Omega_R^i} h_i(u)(u - T_{s_i}) v ds \tag{4.48}$$

The new variational formulation is implemented into FEniCS and simulated iterativly with the applied boundary conditions.

4.10 Summary

The chapter was dedicated to the derivation of the variational formulation, heat generation and boundary conditions for the FEM simulation. The variational formulation was derived for both the one- and three-dimensional case. The source term in the variational form was due to the heat generated in the circuit. The heat generation in the circuit has been calculated for the Mo resistor in the circuit with varying bias current. The heat propagation through the circuit was influenced by the boundary conditions applied to the circuit. The three different boundary conditions and the effects each has on the heat transfer problem has been discussed. The variational formulation, boundary conditions, the source term was combined to solve the heat transfer problem. The Newton's method used to solve the nonlinear PDE was derived. The solver iterates until the approximate relative error of 0.001 was reached.

Chapter 5

Thermal Simulation

5.1 Introduction

This chapter focuses on the implementation of thermal simulation software for multidimensional structures. An overview of the Thermal Analysis tool will be presented in the following section. The heat transfer through the structure is calculated using the PDE solver FEniCS [49]. The simulator imports a meshed multidimensional structure and simulates the heat propagation through the structure, depending on the applied parameters. The simulator solves the heat conduction problems iteratively using the finite element method (FEM). The simulation iterates until the errors between iterations are within the desired range. The simulation outputs the temperature and thermal conductivity for further analysis.

5.2 Design Overview

Figure 5.1 shows the simplified flow diagram of the superconductor thermal analysis simulation tool. The flow diagram is composed of four sections. The input section imports the mesh and parameters for the simulation. The mesh extraction parses the input mesh into the boundary, subdomain and mesh coordinate sections. The preprocessing section prepares the simulation model. The computation section solves the heat transfer problem iteratively. The output section outputs the temperature and thermal conductivity points visually. These sections will be discussed in more detail later in this chapter.

5.3 Input Conditions

The thermal analysis tool, hereafter called Meltdown, operates through a command-line interface (CLI). Figure 5.1 shows the input conditions required

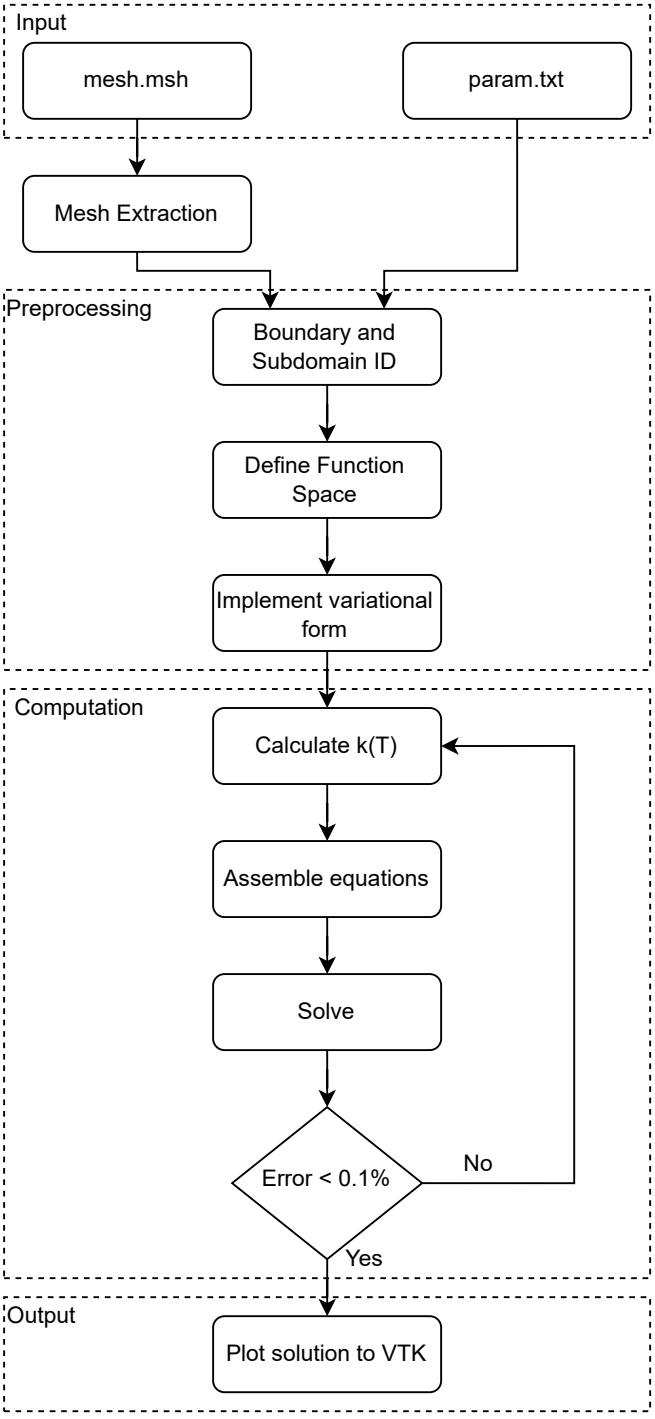


Figure 5.1: The flow diagram of the superconductor thermal analysis simulation tool.

by Meltdown. The two input files required for thermal analysis simulations are the parameter file (.txt) and the mesh file (.msh).

5.3.1 Parameter

The *param.txt* file configures Meltdown by supplying the boundary and testing conditions for the simulation. The three sections of the parameter file are boundary conditions, global parameters, and subdomain information. The boundary conditions for the meshed structure are provided in the boundary section of the parameter file. The surface number, boundary type and constant value are specified in the boundary section of the parameter file. The global parameter section provides modelling parameters applicable to each simulation. This includes the bias current and parameters affecting the thermal conductivity. The subdomain section of the parameter file assigns the different material properties to each layer. The assigned properties overwrite the material type defined for each layer in the *.msh* file.

5.3.2 Mesh

The steady-state heat transfer PDE is solved over the meshed domain of the two or three-dimensional structure. The geometry of the structure is created in GMSH [85] or by using Katana [86,87]. The Katana mesh option is selected by using the parameter '-*k*'. The open-source finite element mesh generator, GMSH, creates the meshed geometry of the test structure. The meshed file contains the marked boundary points, subdomains, and element coordinates of the test structure. The material type and properties for each layer are defined in the *$PhysicalNames* section of the *.msh* file. The type of mesh used as input influences the mesh and boundary processing.

5.4 Mesh Extraction

The input mesh of any structure with more than one dimension is a mixed-cell mesh. The current configuration of FEniCS does not support mixed elements and must be converted to a single-cell mesh format. The mixed-cell mesh is converted to single-cell format by pruning the lower dimensional elements or parsing the mesh. The prune action is not ideal since the process removes the elements that help to identify the boundaries and subdomains of the meshed object. During the mesh extraction process, the *.msh* file is converted to the eXtensible Data Model and Format (XDMF) and Hierarchical Data Format (HDF5) file format, which is compatible with FEniCS. The mesh is parsed into multiple single-cell XDMF files to preserve elements that are used to mark the boundaries and subdomains. Figure 5.2 shows the mesh extraction process.

The multi-cell XDMF file parses into three core single-cell XDMF components required for the simulation. The mesh component includes the mesh and the element coordinates of the geometry. The boundary component contains information concerning the boundary conditions and the physical markings on each boundary. The subdomain component contains the markings of the

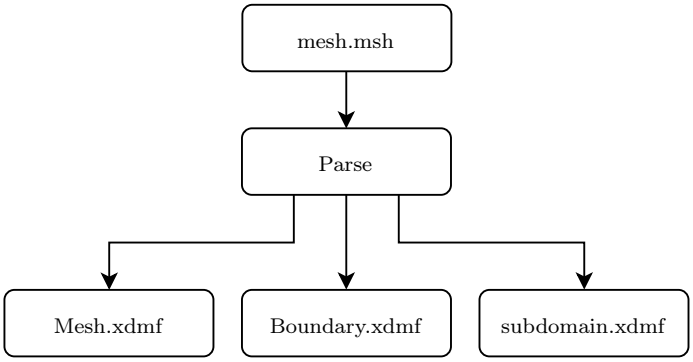


Figure 5.2: Mesh extraction flow diagram. The .msh file of mixed cell type converts to three XDMF meshed objects of single-cell type.

plane surfaces/volume elements. FEniCS imports the XDMF components and constructs the mesh for the FEM problem.

5.5 Boundary Conditions

FEniCS imports the boundary points of the meshed structure from the boundary.xdmf file. The boundary.xdmf file contains the marked facets of the cells that belong to each boundary. The boundary conditions added to the meshed structure is defined in the *param* file. The boundary points match with the boundary conditions, defined in the *param* file. The boundaries on the domain are marked with the boundary type and pass the respective condition to the variational formulation. The last term from (4.10) and (4.17) is the boundary condition calculation for the one and three-dimensional variational form respectively.

5.6 Thermal Conductivity

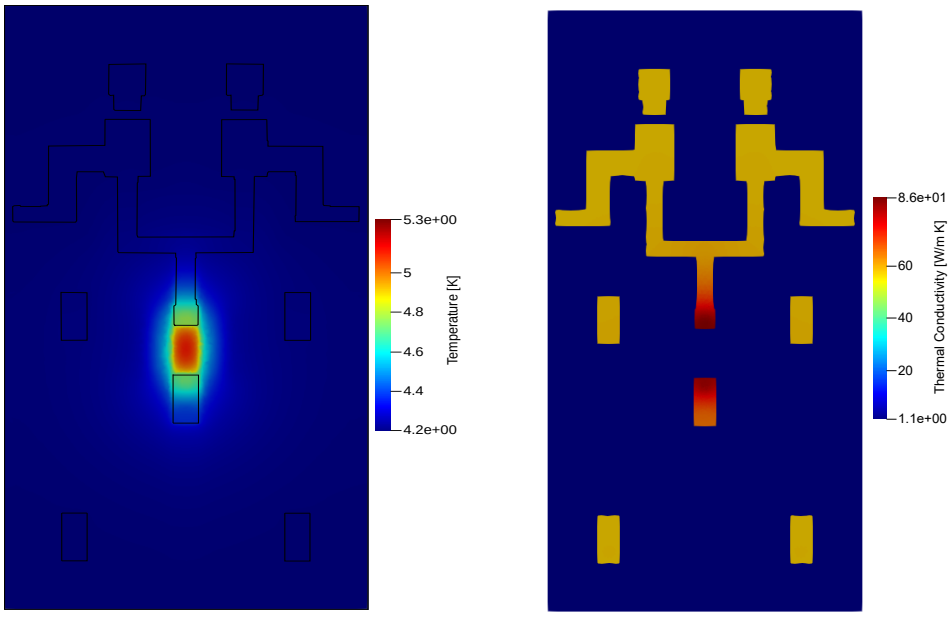
Integrated circuits designed for superconductor electronics are fabricated on a 200mm-diameter Si wafer with all the layers deposited on top. The number of superconducting layers depends on the fabrication process used. The circuit designed using the MIT LL 100 $\mu A/\mu m^2$ SFQ5ee process consists of Superconducting Nb wiring, Mo resistor, SiO_2 dielectric and Nb/AlO_x-Al/Nb JJs layers [3, 6, 36]. The thermal conductivity depends on the material properties of each layer. The meshed object is divided into sections with each subdomain consisting of a different material. The thermal conductivity is calculated for each subdomain to solve the heat transfer PDE.

The subdomains are defined in the *subdomain.xdmf* file, generated by the mesh extraction function. The *subdomain.xdmf* file contains the marked vertex points of the cells in the subdomain. The marked vertices match with the material properties defined either in the *.msh* or *param* file. For each subdo-

main, the thermal conductivity is calculated over each vertex of the cell. The temperature-dependent thermal conductivity is calculated in each subdomain as the simulation proceeds.

5.7 FEM Simulation

The heat transfer through the meshed structure is solved using the FEniCS computing platform [49]. The FEniCS PDE solver imports the multi-dimensional mesh, boundary conditions, source term and material properties. The FEniCS simulates the heat propagation using the variational formulation of the nonlinear heat transfer PDE and the input data. The nonlinear heat transfer PDE is solved iteratively using Newton's method. After each iteration, the temperature-dependent thermal conductivity is recalculated and used in the next iteration. The solver iterates until the solution converges.



(a) Temperature output of the bias resistor.

(b) The thermal conductivity output.

Figure 5.3: The temperature and thermal conductivity output for a JTL submersed in liquid He. A bias current of 2 mA was applied to the resistor.

5.8 Simulation Output

The output section of Figure 5.1 shows the output files to the heat transfer simulation. The heat transfer simulation outputs the temperature and thermal conductivity results in the Visualization Toolkit (VTK) file format for further processing. The VTK file can be opened using an external data analysis and

visualization application such as ParaView [88]. Figure 5.3 shows the temperature and thermal conductivity output for a JTL circuit. The visual output shows the change in temperature and thermal conductivity as the boundary conditions and source are changed. The effect of Joule heating with alternating DC bias current through the bias resistor on the circuit can be studied.

5.9 Summary

The chapter develops the thermal analysis tool to simulate the heat propagation through meshed structures. The heat transfer theory discussed in Chapter 4 was applied into a software solution for the heat transfer problem. The design overview of the simulation process was developed and discussed. The input files provide the meshed structure and parameters applicable to the simulation. The mesh was further processed, and the applicable boundary conditions and thermal conductivity was calculated. The setup of the FEM simulation was discussed along with the results to the simulation.

Chapter 6

Simulation Results

6.1 Introduction

The chapter demonstrates the heat transfer through multidimensional structures using the Meltdown thermal analysis tool. The structures were simulated using a variety of input and boundary conditions at cryogenic temperatures. The simulation shows heat transfer in one, two and three dimensions depending on the input mesh and parameters applied. Meltdown solves the heat transfer problem iteratively until the stopping condition $e_a \leq 0.001$ was reached. The simulations assume inter boundary resistance was negligible and zero magnetic field.

6.2 1D Heat Transfer Example

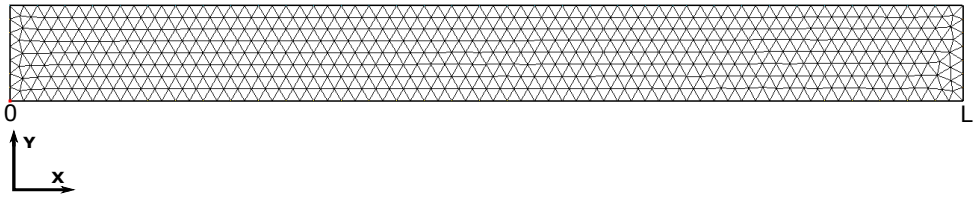
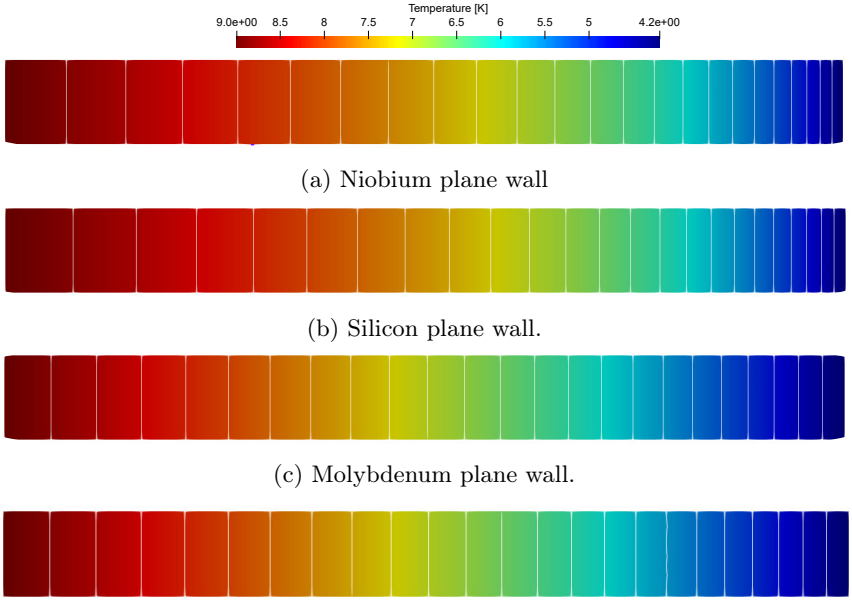


Figure 6.1: The mesh diagram of the plane wall used for the 1D simulations. The length in the x-direction is finite and infinite in the y-direction.

The plane wall simulation shows the influence of temperature-dependent thermal conductivity on the steady-state 1D heat transfer. Figure 6.1 shows the meshed plane wall used for the 1D simulation. The heat propagates only in the finite x-direction by assuming the y-dimension is infinite in length. The one-dimensional heat transfer simulation is treated as a two-point boundary value problem with a length of L = 200 um in the x-direction. The Dirichlet boundary conditions applied at $x(0)$ and $x(\mathrm{L})$ is 4.2K and 9K respectively for

$x \in \Omega = [0, L]$. The heat transfer simulation was performed for plane walls made of niobium, silicon dioxide, molybdenum and silicon.



(a) Niobium plane wall

(b) Silicon plane wall.

(c) Molybdenum plane wall.

(d) Silicon dioxide plane wall.

Figure 6.2: The heat transfer through plane wall for the each material shown. The contour lines are plotted at temperature intervals of 0.2K. The length in the x-direction was set as L = 200 um. The temperature legend is shown on the top of the simulated results.

Figure 6.2 shows the steady-state temperature output for the heat transfer simulation for each of the materials. The contour lines are plotted at temperature intervals of 0.2K. Figure 6.3 shows the temperature distribution through the Nb, Mo, Si and SiO_2 plane walls. The thermal conductivity increases rapidly as the temperature increases for the Nb and Si sample shown in Figure 6.2a and 6.2b respectively. The increased conductivity leads to the heat spreading further through the plane wall until a steady-state is reached. The high rate of heat transfer is shown by the distance between each of the temperature intervals. The larger interval on the $x[0]$ side represents more heat energy transported through the plane wall. The heat transfer through the Mo layer shown in Figure 6.2c is slightly due to the lower thermal conductivity compared to Nb and Si. Figure 6.2d shows the simulated heat transfer through a SiO_2 plane wall. The thermal conductivity of the SiO_2 plane wall was assumed constant, resulting in the linearly spaced out temperature contour lines. Table 6.1 shows the summary of the results for the 1D heat conduction simulations. The results include the material, the number of iterations and the mean temperature value of the output.

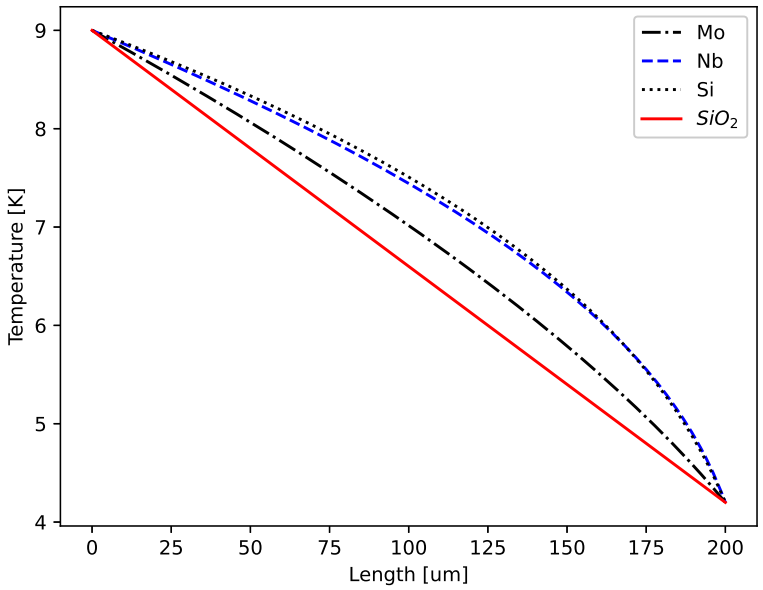


Figure 6.3: The temperature distribution through a Nb, Mo, Si and SiO_2 plane wall.

Material	Iterations	Error	Mean Temp
Nb	11	2.86e-04	7.44K
Si	13	2.43e-04	7.51K
Mo	6	6.71e-04	7.01K
SiO_2	1	0.0	6.6K

Table 6.1: Summary of the results from the 1D plane wall simulations.

6.3 2D Heat Transfer Example

This section simulates the heat conduction through two-dimensional structures. The meshed two-dimensional structure with the applicable boundary conditions are imported to Meltdown and solved iteratively. The simulation demonstrates the propagation of heat flux in both the x- and y-direction. The simulated structures include a Nb block with a heat source in the top center and a Nb layer deposited on Si with a heat source on the top center of the Nb layer.

6.3.1 Heated Strip

The heated strip test structure simulates the steady-state heat transfer through the niobium thin film in 2D. The Mo heating strip was deposited on the Nb thin-film and simulated until steady-state was reached. Figure 6.4 shows the heated strip test structure. The dimension of the thin film and heated strip

was designed as 600 nm x 200 nm and 120 nm x 20 nm respectively. The Mo strip was heated to a temperature of 9K with the bottom on the Nb strip cooled to 4.2K. The Dirichlet boundary temperature of 4.2K approximated the use of liquid helium cooling or a cold finger attached at the bottom of the thin film. The heated strip structures with the applicable boundary conditions were simulated iteratively until the steady-state was reached.

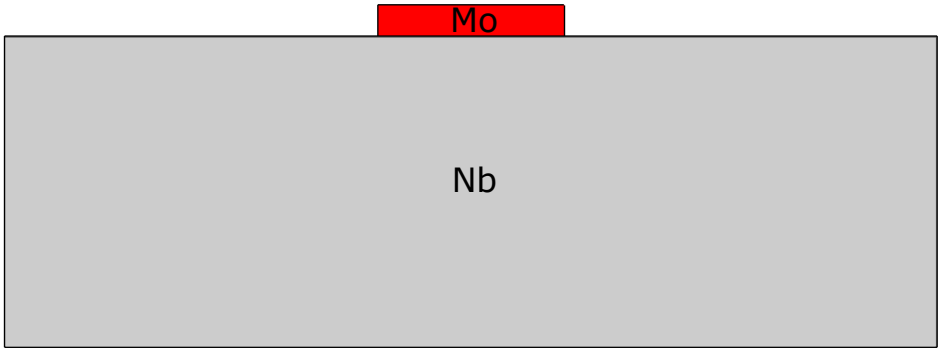


Figure 6.4: The 2D model of the heated strip test structure. The Mo heated strip is deposited on the Nb film.

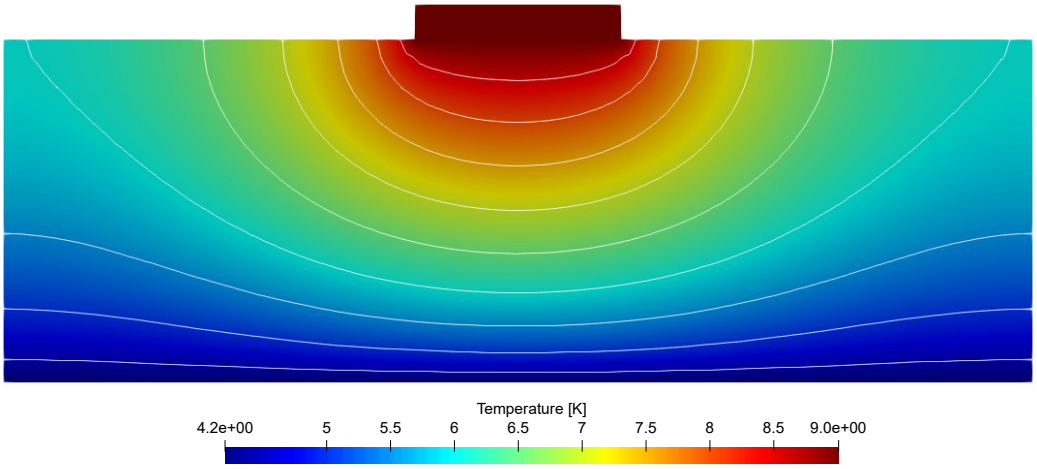


Figure 6.5: The results to the thermal analysis of the thin film structure with the heated strip. The heated strip is kept at a temperature of 9K and the bottom boundary is cooled down to 4.2K. The contour lines showing regions with similar temperature is separated 0.5K apart.

Figure 6.5 shows the steady-state heat conduction of the heated strip structure. The contour lines are plotted at temperature intervals of 0.5K. The figure shows the rapid spherical expansion of heat through the Nb thin film. The rapid spread of temperature was expected from the thermal conductivity calculation in Chapter 3. It was shown that the thermal conductivity of Nb

increases sharply with a small increase in temperature for $T < T_c$. The increased thermal conductivity allows niobium to conduct heat easier through the structure, increasing the rate of heat transfer.

6.3.2 Heated Layered Structure

The heated layer structure simulates the steady-state cross-plane thermal conductivity of the Nb thin film. Figure 6.6 shows the 2D model of the heated layer structure. The Nb thin film (600 nm x 100 nm) was deposited on a Si substrate (600 nm x 200 nm). The Si substrate has a larger thermal conductivity than the Nb thin film. The Mo heated strip (120nm x 20 nm) was deposited on top of the Nb thin film. The surface roughness between the layers was assumed to be negligible. The heated strip was heated to a temperature of 9K. The bottom boundary was cooled to the liquid He temperature, 4.2K. The heated layer structure with the applied boundary conditions was simulated iteratively until the steady state was reached.

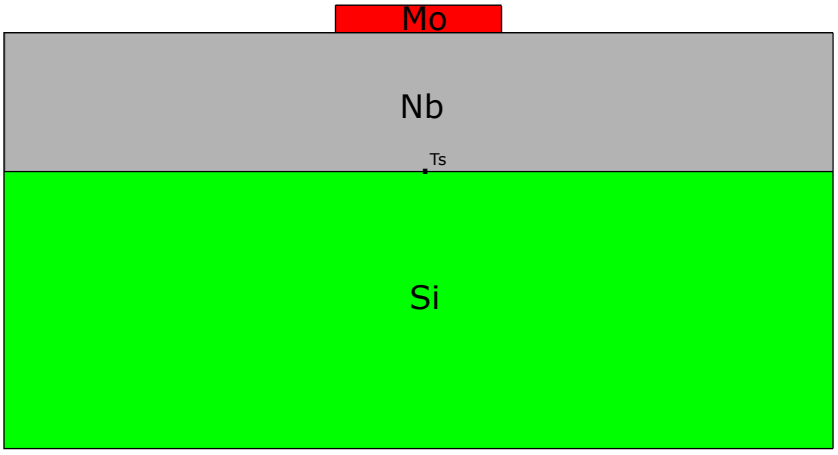
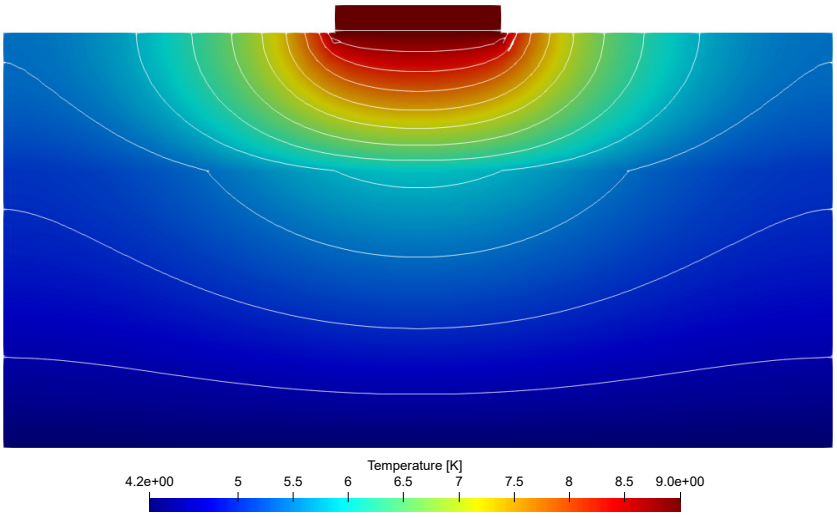
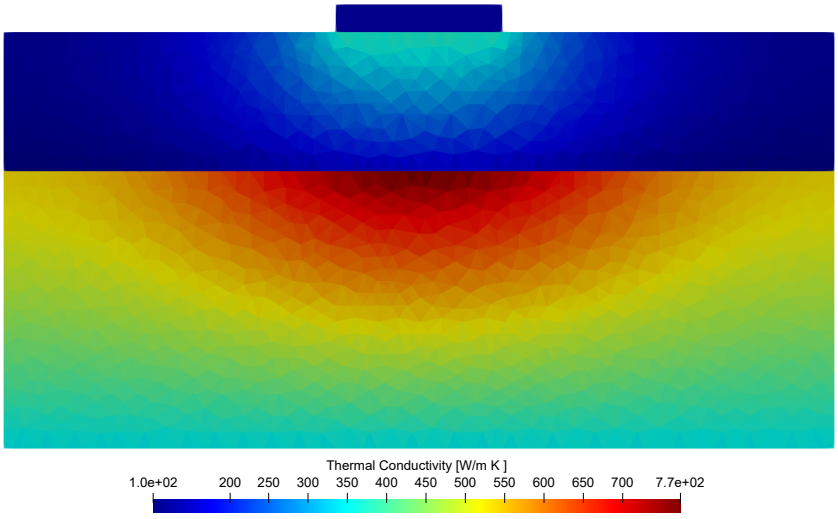


Figure 6.6: The 2D model of the heated layer test structure. The Nb layer is deposited on Si and the Mo heated strip is deposited on the Nb film.

Figure 6.7 shows the steady-state output of the heated dual-layer structure. Figure 6.7a shows the heat transfer through the Nb layer into the Si substrate with contour lines spaced 0.2K apart. Figure 6.7b shows the change in thermal conductivity in both the Nb layer and Si substrate. The heat propagated through the Nb layer at a slower rate than the Si substrate on which it was deposited. The contour lines depicting the temperature change is more densely packed in the Nb layer than the Si substrate. This phenomenon is due to the significant difference in the thermal conductivity of the Nb layer and Si substrate. The larger the thermal conductivity, the larger amount of heat is transferred through the material. Figure 6.7b also shows the change in



(a) Steady-state temperature output for the heated two layer structure. The contour lines are spaced 0.2K apart.

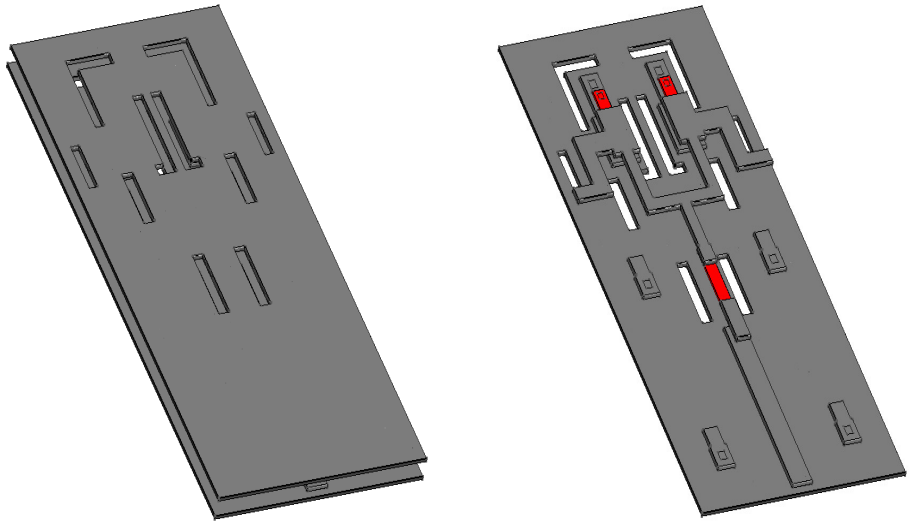


(b) The steady-state thermal conductivity for the heated two layer structure.

Figure 6.7: The steady-state temperature and thermal conductivity output for the heated two layer structure. The temperature of the Mo strip is 9K and the bottom of the Si wafer is kept at a constant 4.2K.

thermal conductivity with temperature for both the Nb and Si layer. The maximum thermal conductivity value for niobium was underneath the heated strip. The thermal conductivity decreased the further away from the resistor it was sampled due to decrease in temperature. The decrease in temperature was due to the lower rate of heat transfer. The experiment was performed to better understand the heat transfer through materials with different thermal conductivity values.

6.4 3D Heat Transfer Example



(a) JTL with top and bottom Nb wiring layers.

(b) JTL with sky plane removed to view resistors and JJs.

Figure 6.8: 3D model of the JTL generated by Katana.

This section simulated the heat propagation and thermal conductivity for a JTL circuit. The JTL from the RSFQ cell library [89] was used for the thermal simulation. Katana creates the JTL mesh, with the boundary and subdomain markers, used in the simulation. Figure 6.8 shows the JTL both with and without the top Nb wire layer in Figure 6.8a and Figure 6.8b, respectively. Grey represents the Nb layers and red represents the Mo resistors. Figure 6.9 shows the JTL circuit with the added dielectric SiO_2 layers deposited on the Si wafer. The yellow and green components represent the SiO_2 dielectric and Si wafer, respectively. The Si wafer was approximated as 0.5mm thick.

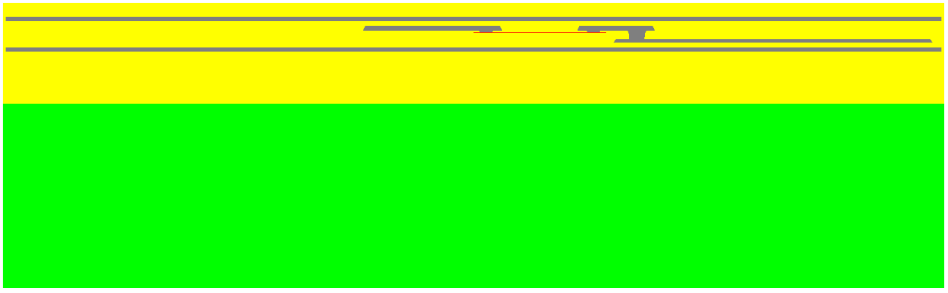


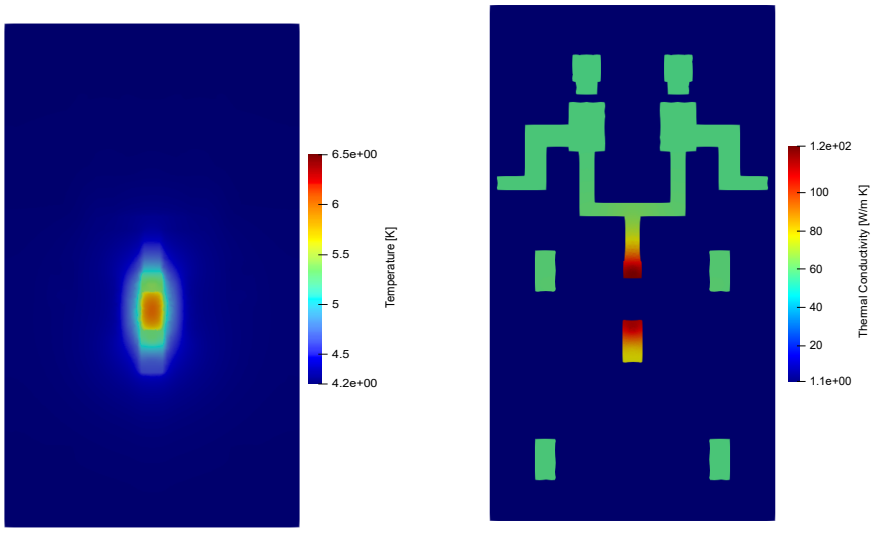
Figure 6.9: JTL with the added dielectric SiO_2 layers deposited on the Si wafer. The yellow and green components represent the SiO_2 dielectric and Si wafer, respectively.

The simulation investigates the influence of the heat generated by the bias resistor on the rest of the circuit. The temperature increase at the JJ and the

Nb vias in contact with the Mo resistor, is of interest. The heat dissipation of the bias resistor depends on the bias current applied. The current through the bias resistor is gradually increased from 0.25 mA to 7.0 mA with intervals of 0.25 mA using the *sweep* function. The temperature point at the vias of the bias resistors, and the temperature close to the JJ is measured over the current sweep. The JTL is simulated for two different testing environments. With the first simulation, the circuit is cooled using a cold finger in contact with the Si wafer in a vacuum. For second simulation, the JTL circuit in a He pool bath. The results of the simulations with each of its effect on the vias and JJs is shown in this section.

6.4.1 Cold Finger

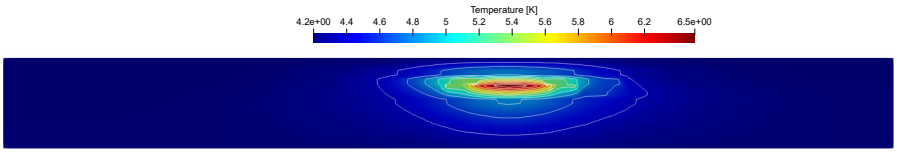
The simulation investigates the heat transfer through the JTL that was cooled using the cold finger method. The 4.2K constant temperature applied to the bottom of the Si wafer approximates the cold finger. Due to the size of the wafer, the inter boundary resistance between the cold finger and Si wafer was negligible. The JTL was simulated in a perfect vacuum environment with only thermal radiation heat transfer applicable to the top of the circuit. The JTL was cooled to a constant temperature of 4.2K before the simulation commenced. This mimics the cooling of the circuit before testing. The emissivity value of $\varepsilon = 0.018$ was used. The simulation was performed using the current sweep and was solved iteratively until the relative error was less than 0.1%. Appendix E shows the command line output for the 3.00mA Cold Finger simulation.



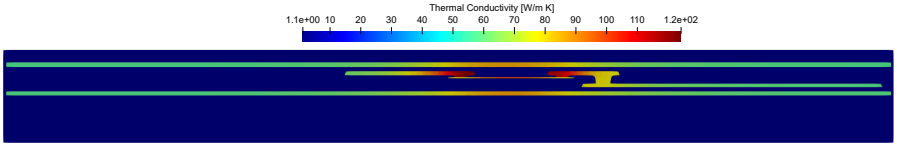
(a) Temperature output of the bias resistor.

(b) The thermal conductivity output of the Nb layers connected to the bias resistor.

Figure 6.10: The top view of the temperature and thermal conductivity for the simulated JTL with a bias current of 3.00mA. The output models were sliced at the vias.



(a) The side view of the temperature output around the bias resistor. The contour lines are spaced 0.2K apart.



(b) The side view of the thermal conductivity output around the bias resistor.

Figure 6.11: The side view of the temperature and thermal conductivity for the simulated JTL with a bias current of 3.00mA. The output models were sliced through the center of the bias resistor.

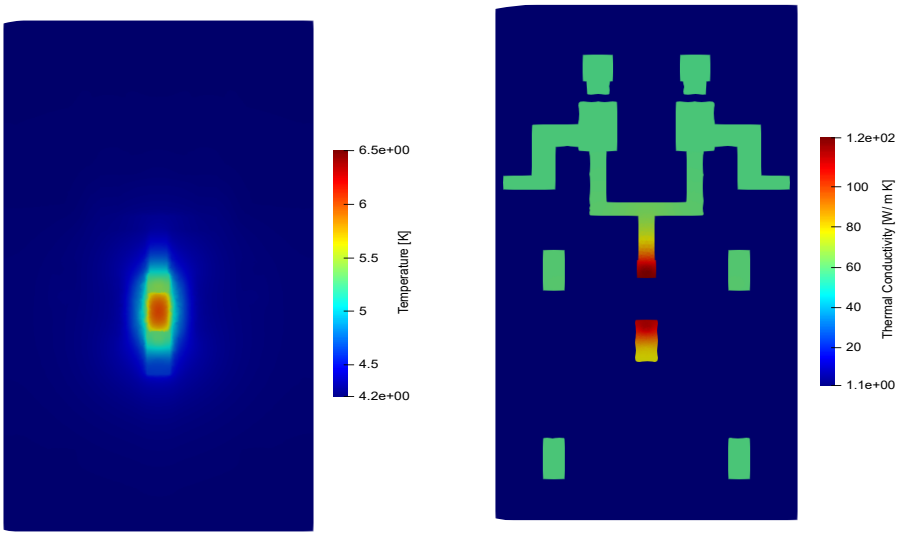
Figure 6.10 and Figure 6.11 shows the temperature and thermal conductivity output for the JTL cold finger simulation using a bias current of 3.00mA. Figure 6.10 displays the top view of the temperature and thermal conductivity output sliced at the vias parallel to the bias resistor. Figure 6.11 shows the temperature and thermal conductivity output through the bias resistor and the connecting wiring layers, sliced perpendicular to the bias resistor. Figure 6.10b and 6.11b shows the change in thermal conductivity of the Nb layers with an increase in temperature. The thermal conductivity of Nb increased significantly around the bias resistor to the connected Nb layer as the heat propagates through the circuit. The increased thermal conductivity increases the heat transfer rate in that region, resulting in the increased temperature of the Nb layers near the vias. The increase in thermal conductivity of Nb with temperature agrees with what was derived in Chapter 3.

The largest temperature change was present in the localised area close to the resistor due to the increased thermal conductivity. The limited propagation of heat through the Nb layer is shown in Figure 6.10a. The increase in temperature was limited to the vias of the bias resistor. Figure 6.11a shows the heat transfer from the bias resistor to the vias, which is connecting the Nb layer to the bias resistor. The heat dissipated by the bias resistor was mostly contained to the region between top and bottom Nb wiring layers. The contour lines on the temperature diagram are spaced 0.2K apart. The extensions on the sides of the contour lines are the heat transmitted by the Nb layer connected to the bias resistor.

6.4.2 Helium Pool Boiling

The simulation investigates the heat transfer through a JTL submerged in liquid helium. The heat dissipated by the bias resistor is transferred to the

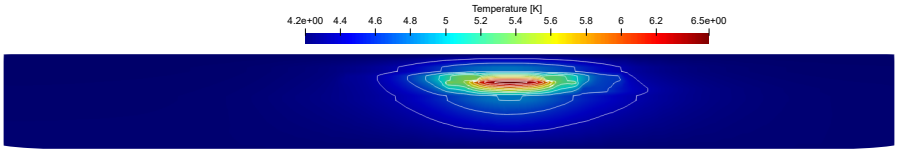
surrounding liquid helium by making use of pool boiling heat transfer. The liquid helium boundary was applied to the top of the circuit and the bottom of the Si wafer. Natural boundary conditions were applied to the sides to simulate the JTL as a small section of a large complete circuit. The heat transfer due to thermal radiation was negligible compared to the convective heat transfer of the liquid helium. The JTL was cooled to a constant temperature of 4.2K before the simulation commenced. This mimics the cooling of the circuit before testing. The simulation was performed using the current sweep and was solved iteratively until the relative error was less than 0.1%.



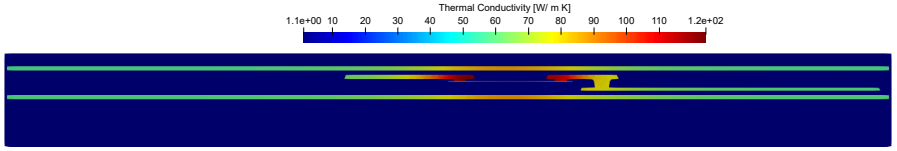
(a) Temperature output of the bias resistor. (b) The thermal conductivity output of the Nb layers connected to the bias resistor.

Figure 6.12: The top view of the temperature and thermal conductivity for the simulated JTL with a bias current of 3.00mA. The output models were sliced at the vias.

Figure 6.12 and Figure 6.13 show the temperature and thermal conductivity output for the JTL helium bath simulation using a bias current of 3.00mA. Figure 6.12 displays the top view of the temperature and thermal conductivity output sliced at the vias parallel to the bias resistor. Figure 6.13 shows the temperature and thermal conductivity output of the bias resistor and the connecting wiring layers, sliced perpendicular to the bias resistor. The helium pool bath simulation results are similar to the cold finger output. The temperature at the top boundary increased a fraction of a degree when the steady-state was reached. The slight increase in temperature past the boiling point may cause the helium on the surface to form very small bubbles. The temperature on the surface must increase further before nucleating boiling occurs.



(a) The side view of the temperature output around the bias resistor. The contour lines are spaced 0.2K apart.



(b) The side view of the thermal conductivity output around the bias resistor.

Figure 6.13: The side view of the temperature and thermal conductivity for the simulated JTL with a bias current of 3.00mA. The output models were sliced through the center of the bias resistor.

6.4.3 Via Temperature

The temperature of the vias connecting the bias resistor to the wiring layer was simulated for the current sweep. Figure 6.14 shows the zoomed-in bias resistor section of the layout with the numbered vias temperature point. The heat transfer through the JTL was simulated until the temperature at one of the two vias points reached their critical temperature. If the temperature increases past the critical temperature point, the Nb wire layer in contact with the bias resistor will transition back into the normal state.

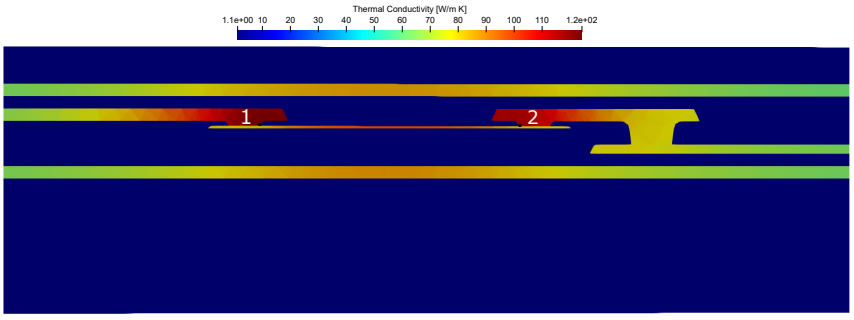
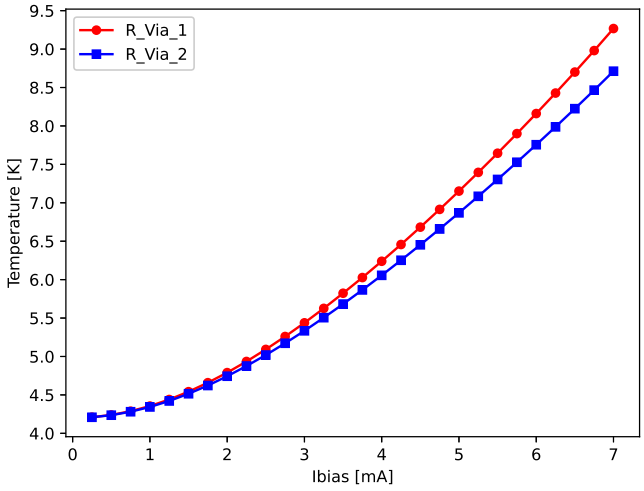


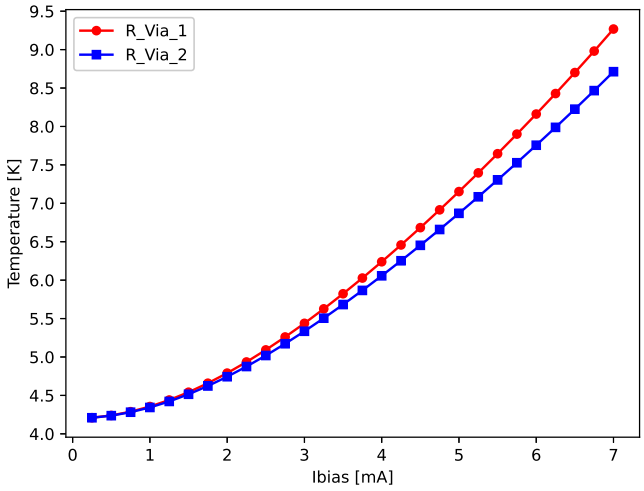
Figure 6.14: The zoomed in section of the thermal conductivity output of the JTL marked with the via numbers and maximum temperature point.

Figure 6.15 shows the plots of the temperature versus current at both the vias of the resistor during the simulation. Figure 6.15a and 6.15b shows the plot for the cold finger and pool boiling simulation, respectively. From Ibais = 2.50mA, the temperature of the R_Via_1 increase at a greater rate than R_Via_2. The temperature difference may be due to the layer interconnect on the input of the JTL, as shown in Figure 6.14. The layer interconnect is

filled with niobium metal that provides a larger area to heat up, thus reducing the temperature in the *R_ Via_ 2* region. Figure 6.11a and 6.13a shows the heated layer interconnect for both the cold finger and He pool boiling method. The temperature at Resistor Via 1 reaches a temperature of 9.26K for a bias current of 7.00 mA. At that point, the temperature increased past T_c and the Via exits the superconductive state.



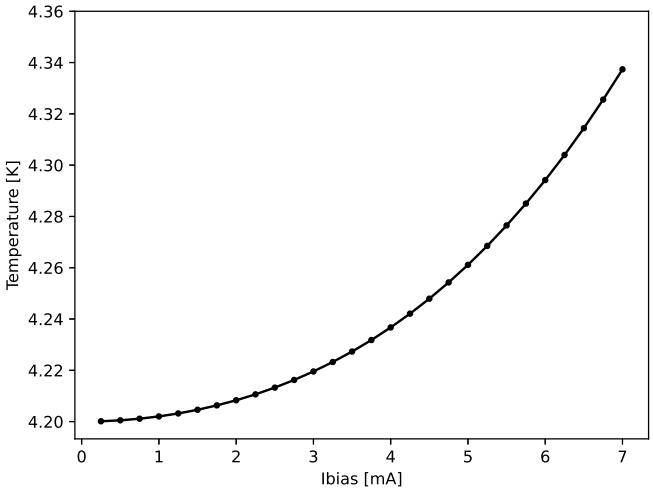
(a) Cold finger via temperature vs. bias current.



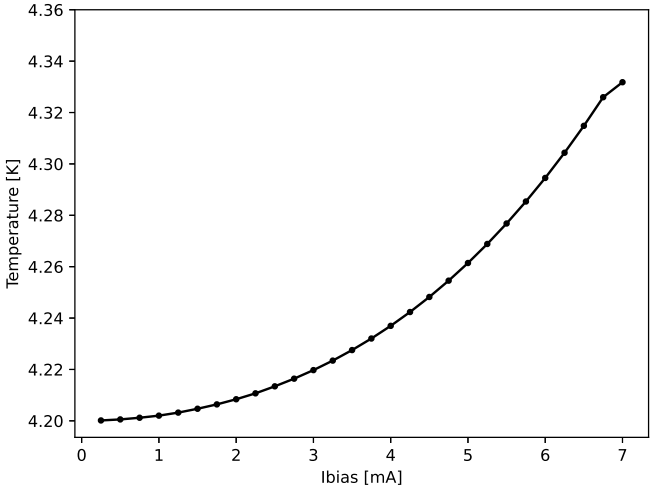
(b) He pool boiling via temperature output vs. bias current.

Figure 6.15: Simulated temperature versus bias current for the vias connecting the Nb wiring layer to the bias resistor.

6.4.4 JJ Temperature



(a) Cold finger JJ temperature vs current simulation.



(b) He pool boiling JJ temperature vs current simulation.

Figure 6.16: Simulated junction temperature with a change in bias current for the cold finger and pool boiling simulation.

The section simulates the temperature increase of the Josephson junction due to the heat dissipated by the bias resistor. Figure 6.16 shows the temperature versus bias current plots of the Josephson junction. Figure 6.16a and 6.16b shows the plot of the simulated Josephson junction temperature for both the cold finger and He pool bath simulation, respectively. The Josephson junction temperature for the cold finger and He pool bath only differs by a

few milliKelvin. The simulation showed that the temperature increase of the Josephson junction due to the bias resistor was minimal for the simulated JTL. With the maximum bias current of 7.00mA, the temperature of the Josephson junction increased by 0.14K. The distance between the bias resistor and the Josephson Junction was large enough to minimise the temperature increase. The temperature change would have been more pronounced if the resistor was closer to the Josephson Junctions

6.5 Summary

The chapter demonstrates the operation of the Meltdown thermal analysis tool. The simulations showed heat propagation through multidimensional structures. The 1D heat transfer section illustrated how the temperature-dependent thermal conductivity affected heat propagation through a 1D plane. The sizeable thermal conductivity on the heated side allowed a greater heat transfer rate which decreased the closer we get to the cooled side. The 2D heat transfer section showed how heat propagated in two dimensions, away from the heated source in a circular fashion. The double layer structure showed the difference in heat spread through layers with different thermal conductivity properties.

The JTL section finally showed the heat transfer due to the heat dissipated by the bias resistor for a range of applied current. The results showed that the heat dissipated by the bias resistor had minimal effect on the JJs due to the distance between the components. The circuit failed at the resistor vias before the temperature in the JJs increased significantly.

It was found that the largest increase in temperature remained localised around the heat source due to the low thermal conductivity of the surrounding dielectric. The low thermal conductivity of the dielectric reduces the amount of heat dissipating out of the circuit, resulting in the increased temperature observed around the source.

Chapter 7

Conclusions and Recommendations

7.1 Conclusion

A thermal analysis tool has been developed to simulate heat transfer through multidimensional superconductor structures. The thermal analysis tool, Meltdown, allows circuit designers to investigate the heat spread as a result of different conditions and input bias currents. The heat transfer simulation can help identify potential problems before fabrication, streamlining the design process.

The temperature-dependent thermal conductivity of various samples at cryogenic temperatures was visualised. The change in thermal conductivity has been proven to influence heat propagation through a structure. Chapter 3 derives the thermal conductivity for superconducting metals at cryogenic temperatures. In the superconductive state, the thermal conductivity is affected by the Cooper pair formation and dissolution. This leads to a change in the electron contribution of the thermal conductivity as the temperature changes. The 1D and 2D thermal simulations demonstrated the influence of temperature-dependent thermal conductivity and heat transfer through a layered structure.

The thermal analysis tool has been able to determine the steady-state heat transfer through three-dimensional layered structures of dissimilar materials. The JTL heat transfer simulation showed the heat dissipation of the bias resistor to the surroundings. The temperature increase was localised around the resistor with minimal heat transfer to the Josephson junctions. The temperature increase of the Josephson junction from only the heat dissipated by the bias resistor was negligible for the present JTL design. If the temperature increase was above the acceptable levels, the problem could have been identified and rectified before fabrication. The assumption of perfect thermal contact between layers and zero magnetic field limits the scope of the developed simulation tool. Meltdown outputs the temperature and thermal conductivity data

to a VTK file. The output data can be further processed by making use of an external data analysis and visualizationa program such as Paraview.

7.2 Recommendations and Future Work

The thermal simulations provide an initial look into heat transfer through superconductor circuits. The thermal analysis tool has the potential for future improvements and broader applications to the superconductor IC design process. Future work may build up upon the thermal model to include the influence of the magnetic field. It was mentioned how the magnitude and direction of the magnetic fields influence the thermal conductivity of Type II superconductors in the intermediate state. The additional factor will improve the thermal model by not only investigating the effect of heat conduction but also the magnetic field on the circuit.

The second recommendation will be to incorporate the thermal contact resistance between the layers of the superconductor circuit. It would be difficult to estimate the temperature drop between the layers and a full measurement of the thermal effects on the circuit is required to simulate the heat propagation more accurately. Measured temperature data will be beneficial especially for thermal contact resistance.

It is recommended to use the thermal analysis tool in conjunction with Katana.

List of References

[1] NIST, "cryogenic material properties Molybdenum." [Online]. Available: https://trc.nist.gov/cryogenics/materials/Molybdenum/Molybdenum{_}rev.htm

[2] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, L. M. Johnson, M. A. Gouker, and W. D. Oliver, "Fabrication process and properties of fully-planarized deep-submicron Nb/Al-AlOx/Nb Josephson junctions for VLSI circuits," *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 3, jun 2015.

[3] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, A. Wynn, D. E. Oates, L. M. Johnson, and M. A. Gouker, "Advanced fabrication processes for superconducting very large-scale integrated circuits," *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 3, 2016.

[4] S. K. Tolpygo, V. Bolkhovsky, S. Zarr, T. J. Weir, A. Wynn, A. L. Day, L. M. Johnson, and M. A. Gouker, "Properties of Unshunted and Resistively Shunted Nb/AlOx-Al/Nb Josephson Junctions With Critical Current Densities From 0.1 to 1 mA/μm2," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, jun 2017.

[5] S. K. Tolpygo, V. Bolkhovsky, D. E. Oates, R. Rastogi, S. Zarr, A. L. Day, T. J. Weir, A. Wynn, and L. M. Johnson, "Superconductor Electronics Fabrication Process with MoNx Kinetic Inductors and Self-Shunted Josephson Junctions," *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 4, 2018.

[6] S. K. Tolpygo, V. Bolkhovsky, R. Rastogi, S. Zarr, A. L. Day, E. Golden, T. J. Weir, A. Wynn, and L. M. Johnson, "Advanced Fabrication Processes for Superconductor Electronics: Current Status and New Developments," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, 2019.

[7] V. K. Semenov, Y. A. Polyakov, and S. K. Tolpygo, "New AC-powered SFQ digital circuits," *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 3, jun 2015.

[8] ——, "AC-Biased Shift Registers as Fabrication Process Benchmark Circuits and Flux Trapping Diagnostic Tool," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, jun 2017.

[9] S. K. Tolpygo and V. K. Semenov, "Increasing integration scale of superconductor electronics beyond one million Josephson junctions," in *Journal of Physics: Conference Series*, vol. 1559, no. 1. Institute of Physics Publishing, jun 2020, p. 12002.

[10] V. Ambegaokar and B. I. Halperin, "Voltage due to thermal noise in the dc Josephson effect," *Physical Review Letters*, vol. 22, no. 25, pp. 1364–1366, 1969.

[11] T. Ortlepp and H. F. Uhlmann, "Noise analysis for intrinsic and external shunted josephson junctions," in *Superconductor Science and Technology*, 2004.

[12] T. Ortlepp and F. H. Uhlmann, "Noise induced timing jitter: A general restriction for high speed RSFQ devices," in *IEEE Transactions on Applied Superconductivity*, vol. 15, no. 2 PART I, jun 2005, pp. 344–347.

[13] S. K. Tolpygo, "Superconductor digital electronics: Scalability and energy efficiency issues," pp. 361–379, may 2016.

[14] T. P. Orlando and K. A. Delin, *Foundations of applied superconductivity*. Addison-Wesley Reading, MA, 1991, vol. 8.

[15] H. K. Onnes, "Investigations into the properties of substances at low temperatures, which have led, amongst other things, to the preparation of liquid helium," *Nobel lecture*, vol. 4, pp. 306–336, 1913.

[16] B. Seeber, *Handbook of applied superconductivity*, B. Seeber, Ed. Bristol: Institute of Physics, 1998.

[17] A. A. Abrikosov, "On the magnetic properties of superconductors of the second group," *Sov. Phys. JETP*, vol. 5, pp. 1174–1182, 1957.

[18] M. Tinkham, *Introduction to superconductivity*, 2nd ed. Mineola, N.Y.: Dover Publications, 2004.

[19] W. Buckel, *Superconductivity : fundamentals and applications.*, 2nd ed. Weinheim: Wiley-VCH, 2004.

[20] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Microscopic Theory of Superconductivity," *Phys. Rev.*, vol. 106, no. 1, pp. 162–164, 1957.

[21] J. Bardeen, G. Rickayzen, and L. Tewordt, "Theory of the thermal conductivity of superconductors," *Physical Review*, vol. 113, no. 4, pp. 982–994, 1959.

[22] u. F. by A Lindemann, "The electromagnetic equations of the supraconductor," *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, vol. 149, no. 866, pp. 71–88, mar 1935.

[23] C. J. Gorter and H. Casimir, "The thermodynamics of the superconducting state," *Z. tech. Phys*, vol. 15, pp. 539–42, 1934.

[24] Y. A. Cengel, *Heat transfer : a practical approach*, 2nd ed., ser. Mcgraw-Hill series in Mechanical Engineering. Boston: McGraw-Hill, 2003.

[25] K. Mendelssohn, "Thermal conductivity of superconductors," *Physica*, vol. 19, no. 1-12, pp. 775–787, jan 1953.

[26] J. L. Olsen and H. M. Rosenberg, "The thermal conductivity of metals at low temperatures," *Advances in Physics*, vol. 2, no. 5, pp. 28–66, 1953.

[27] E. Lynton, *Superconductivity*, 2nd ed. Methuen, 1964.

[28] K. Maki, "Thermal conductivity of pure type-II superconductors in high magnetic fields," *Physical Review*, vol. 158, no. 2, pp. 397–399, 1967.

[29] J. Lowell and J. B. Sousa, "Mixed-state thermal conductivity of type II superconductors," *Journal of Low Temperature Physics*, vol. 3, no. 1, pp. 65–87, jul 1970.

[30] P. H. Kes, J. G. Rolfes, and D. de Klerk, "Thermal conductivity of niobium in the purely superconducting and normal states," *Journal of Low Temperature Physics*, vol. 17, no. 3-4, pp. 341–364, nov 1974.

[31] J. K. Hulm, "The thermal conductivity of tin, mercury, indium and tantalum at liquid helium temperatures," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 204, no. 1076, pp. 98–123, nov 1950.

[32] P. Klemens, "Thermal conductivity of solids at low temperatures," in *Low Temperature Physics I/Kältephysik I.* Springer, 1956, pp. 198–281.

[33] A. Calverley, K. Menclelssohn, and P. M. Rowell, "Some thermal and magnetic properties of tantalum, niobium, and vanadium at helium temperatures," *Cryogenics*, vol. 2, no. 1, pp. 26–33, sep 1961.

[34] C. Kothandaraman, *Fundamentals of heat and mass transfer*, 3rd ed. New Age International, 2006.

[35] Rohsenow, Hartnett, and Cho, *Handbook of heat transfer*, 3rd ed., ser. McGraw-Hill handbooks, W. M. Rohsenow, J. P. J. P. Hartnett, and Y. I. Cho, Eds. New York: McGraw-Hill, 1998.

[36] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, M. Hidaka, H. Akaike, A. Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa, "New nb multilayer fabrication process for large-scale sfq circuits," *Physica C: Superconductivity and its applications*, vol. 469, no. 15, pp. 1578–1584, 2009.

[37] F. Pobell, *Matter and methods at low temperatures*, 3rd ed. Springer International Publishing, 2007.

[38] M. G. Cooper, B. B. Mikic, and M. M. Yovanovich, "Thermal contact conductance," *International Journal of Heat and Mass Transfer*, vol. 12, no. 3, pp. 279–300, 1969.

[39] M. M. Yovanovich, "Four decades of research on thermal contact, gap, and joint resistance in microelectronics," pp. 182–206, 2005.

[40] E. T. Swartz and R. O. Pohl, "Thermal resistance at interfaces," *Applied Physics Letters*, vol. 51, no. 26, pp. 2200–2202, 1987.

[41] G. L. Pollack, "Kapitza Resistance," *Reviews of Modern Physics*, vol. 41, no. 1, pp. 48–81, 1969.

[42] E. T. Swartz and R. O. Pohl, "Thermal boundary resistance," *Reviews of Modern Physics*, vol. 61, no. 3, pp. 605–668, 1989.

[43] A. F. Andreev, "On the thermal conductivity of the intermediate state of a superconductor," *Journal of Physics C: Solid State Physics*, vol. 1, no. 3, pp. 744–747, 1968.

[44] SimScale, "What is FEA | Finite Element Analysis? - SimScale Documentation," p. 1, 2019. [Online]. Available: https://www.simscale.com/docs/simwiki/fea-finite-element-analysis/what-is-fea-finite-element-analysis/

[45] D. L. Logan, *A first course in the finite element method*. Cengage Learning, 2011.

[46] D. W. Pepper and J. C. Heinrich, *The finite element method: basic concepts and applications with MATLAB, MAPLE, and COMSOL*, 3rd ed. CRC press, 2017.

[47] H. P. Langtangen and A. Logg, *Solving PDEs in Python*. Springer International Publishing, 2016.

[48] A. Logg, K.-A. Mardal, G. N. Wells *et al.*, *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[49] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The fenics project version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[50] "MFEM - Finite Element Discretization Library." [Online]. Available: https://mfem.org/

[51] "mfem/mfem: Lightweight, general, scalable C++ library for finite element methods." [Online]. Available: https://github.com/mfem/mfem

[52] "FreeFEM - An open-source PDE Solver using the Finite Element Method." [Online]. Available: https://freefem.org/

[53] "The deal.II Finite Element Library." [Online]. Available: https://www.dealii.org/

[54] "dealii." [Online]. Available: https://github.com/dealii

[55] R. E. Makinson, "The thermal conductivity of metals," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 34, no. 3, pp. 474–497, 1938.

[56] A. Wilson, "The theory of metals cambridge," *Gt. Britain*, p. 26, 1953.

[57] J. D. Splett, D. F. Vecchia, and L. Goodrich, "A Comparison of Methods for Computing the Residual Resistivity Ratio of High-Purity Niobium," *Journal of Research of the National Institute of Standards and Technology*, vol. 116, no. 1, p. 489, 2011.

[58] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of Superconductivity," *Physical Review*, vol. 108, no. 5, pp. 1175–1204, dec 1957.

[59] F. Koechlin, "Parametrization of the niobium thermal conductivity in the superconducting state," *Superconductor Science and Technology*, vol. 9, no. 6, pp. 453–461, June 1996.

[60] F. Simon, F. Lange, R. C. Swain, R. W. B. Stephens, K. R. Wilkinson, and J. Wilks, "The thermal conductivity of dielectric solids at low temperatures (Theoretical)," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 208, no. 1092, pp. 108–133, aug 1951.

[61] K. Mendelssohn and H. M. Rosenberg, "The Thermal Conductivity of Metals at Low Temperatures," *Solid State Physics - Advances in Research and Applications*, vol. 12, no. C, pp. 223–274, jan 1961.

[62] H. B. Casimir, "Note on the conduction of heat in crystals," *Physica*, vol. 5, no. 6, pp. 495–500, 1938.

[63] S. K. Chandrasekaran, N. T. Wright, T. R. Bieler, and C. C. Compton, "Effect of Heat Treatment Temperature on the Thermal Conductivity of Large Grain Superconducting Niobium," in *Proceedings of the SRF2011*, Chicago, 2011, pp. 593–596.

[64] M. Graf, S. K. Yip, J. Sauls, and D. Rainer, "Electronic thermal conductivity and the Wiedemann-Franz law for unconventional superconductors," *Physical Review B - Condensed Matter and Materials Physics*, vol. 53, no. 22, pp. 15 147–15 161, 1996.

[65] S. M. Wasim and N. H. Zebouni, "Thermal conductivity of superconducting niobium," *Physical Review*, vol. 187, no. 2, pp. 539–548, 1969.

[66] S. K. Chandrasekaran, T. R. Bieler, C. Compton, and N. T. Wright, "Phonon scattering in the thermal conductivity of large-grain superconducting niobium as a function of heat treatment temperature," in *AIP Conference Proceedings*, vol. 1434, no. 57, 2012, pp. 976–982.

[67] W. Wasserbäh, "Low-temperature thermal conductivity of plastically deformed niobium single crystals," *Philosophical Magazine A: Physics of Condensed Matter, Structure, Defects and Mechanical Properties*, vol. 38, no. 4, pp. 401–431, 1978.

[68] T. Yamane, N. Nagai, S. I. Katayama, and M. Todoki, "Measurement of thermal conductivity of silicon dioxide thin films using a 3ω method," *Journal of Applied Physics*, vol. 91, no. 12, pp. 9772–9776, jun 2002.

[69] H. C. Chien, D. J. Yao, M. J. Huang, and T. Y. Chang, "Thermal conductivity measurement and interface thermal resistance estimation using SiO 2 thin film," *Review of Scientific Instruments*, vol. 79, no. 5, 2008.

[70] H. M. Rosenberg, "The Thermal Conductivity of Metals at Low Temperatures," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 247, no. 933, pp. 441–497, mar 1955.

[71] K. Mendelssohn and H. M. Rosenberg, "The thermal conductivity of metals at low temperatures II: The transition elements," *Proceedings of the Physical Society, Section A*, vol. 65, no. 6, pp. 388–394, 1952.

[72] C. J. Glassbrenner and G. A. Slack, "Thermal conductivity of silicon and germanium from 3K to the melting point," *Physical Review*, vol. 134, no. 4A, p. A1058, may 1964.

[73] M. G. Holland, "Analysis of lattice thermal conductivity," *Physical Review*, vol. 132, no. 6, pp. 2461–2471, dec 1963.

[74] C. Y. Ho, R. W. Powell, and P. E. Liley, "Thermal Conductivity of the Elements," *Journal of Physical and Chemical Reference Data*, vol. 1, no. 2, pp. 279–421, apr 1972.

[75] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, ser. Texts in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 10.

[76] H. P. Langtangen and K.-A. Mardal, "Introduction to numerical methods for variational problems," 2016.

[77] E. E. Kalu *et al.*, *Numerical Methods with Applications: Abridged*. Lulu. com, 2009.

[78] R. V. Smith, "Review of heat transfer to helium I," vol. 9, no. 1, pp. 11–19, feb 1969.

[79] C. Johannes, "Recent advances in heat transfer to helium 1," *Revue de Physique Appliquée*, vol. 6, no. 4, pp. 509–513, dec 1971.

[80] B. Baudouy, "Heat Transfer and Cooling Techniques at Low Temperature," *CAS-CERN Accelerator School: Superconductivity for Accelerators - Proceedings*, pp. 329–352, jan 2015.

[81] S. W. Van Sciver, *Helium cryogenics: Second edition*. Springer Science & Business Media, 2012.

[82] S. Kutateladze, "Heat transfer in condensation and boiling, state sci. & tech," *Pub. of Lit on Machinerary, Moskau (Atomic Energy Commission Translation 3770, Tech. Info. Service, Oak Rich Tennesee)*, 1952.

[83] E. Brentari, *Boiling heat transfer for oxygen, nitrogen, hydrogen, and helium*. US National Bureau of Standards, 1965, vol. 13.

[84] N. Zuber, "The hydrodynamic crisis in pool boiling of saturated and subcooled liquids," *Int. Developments in Heat Transfer, ASME*, vol. 27, pp. 230–236, 1961.

[85] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, sep 2009.

[86] H. F. Herbst, "Gate-Level Superconductor Integrated Circuit Fabrication Process Modelling for Improved Layout Extraction, MEng Thesis, Stellenbosch University," 2020.

[87] H. F. Herbst, P. L. Roux, K. Jackman, and C. J. Fourie, "Improved Transmission Line Parameter Calculation through TCAD Process Modeling for Superconductor Integrated Circuit Interconnects," *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 7, oct 2020.

[88] "ParaView." [Online]. Available: https://www.paraview.org/

[89] L. Schindler, "RSFQ cell library V1.5.1." [Online]. Available: https://github.com/sunmagnetics/RSFQlib

[90] C. J. Fourie, *InductEx User Manual*, 5th ed., Stellenbosch University.

[91] "WRspice Reference Manual," Tech. Rep., 2019.

[92] C. J. Fourie, K. Jackman, M. M. Botha, S. Razmkhah, P. Febvre, C. L. Ayala, Q. Xu, N. Yoshikawa, E. Patrick, M. Law, Y. Wang, M. Annavaram, P. Beerel, S. Gupta, S. Nazarian, and M. Pedram, "Cold-Flux Superconducting EDA and TCAD Tools Project: Overview and Progress," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, 2019.

[93] C. Fourie, "Single flux quantum circuit technology and CAD overview," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*. Institute of Electrical and Electronics Engineers Inc., nov 2018.

[94] S. Razmkhah and P. Febvre, "JOINUS: A User-Friendly Open-Source Software to Simulate Digital Superconductor Circuits," *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 5, aug 2020.

[95] P. L. Roux, K. Jackman, J. A. Delport, and C. J. Fourie, "Modeling of Superconducting Passive Transmission Lines," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, aug 2019.

[96] C. J. Fourie and K. Jackman, "Software tools for flux trapping and magnetic field analysis in superconducting circuits," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, aug 2019.

[97] J. A. Delport, K. Jackman, P. L. Roux, and C. J. Fourie, "JoSIM - Superconductor SPICE simulator," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, aug 2019.

[98] "JoSIM v2.3 Documentation." [Online]. Available: https://joeydelp.github.io/JoSIM/

[99] K. K. Likharev and V. K. Semenov, "RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-Terahertz-Clock-Frequency Digital Systems," *IEEE Transactions on Applied Superconductivity*, vol. 1, no. 1, pp. 3–28, 1991.

[100] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Superconductor Science and Technology*, vol. 26, no. 3, mar 2013.

[101] K. Inoue, N. Takeuchi, K. Ehara, Y. Yamanashi, and N. Yoshikawa, "Simulation and experimental demonstration of logic circuits using an ultra-low-power adiabatic quantum-flux-parametron," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, 2013.

[102] N. Takeuchi, S. Nagasawa, F. China, T. Ando, M. Hidaka, Y. Yamanashi, and N. Yoshikawa, "Adiabatic quantum-flux-parametron cell library designed using a 10 kA cm-2 niobium fabrication process," *Superconductor Science and Technology*, vol. 30, no. 3, mar 2017.

[103] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Measurement of 10 zJ energy dissipation of adiabatic quantum-flux- parametron logic using a superconducting resonator," *Applied Physics Letters*, vol. 102, no. 5, feb 2013.

[104] ——, "Adiabatic quantum-flux-parametron cell library adopting minimalist design," *Journal of Applied Physics*, vol. 117, no. 17, may 2015.

[105] L. C. Muller and C. J. Fourie, "Automated state machine and timing characteristic extraction for RSFQ Circuits," *IEEE Transactions on Applied Superconductivity*, vol. 24, no. 1, 2014.

Appendices

Appendix A

XIC Integration

A.1 Introduction

The superconductor circuit design process involves alternating between multiple software packages for creation, testing and verification of circuit layouts. The software packages required for this process include XicTools, JoSIM and InductEx.

The current process involves parsing the data extracted from the software packages. The data is processed manually and reformatted for compatibility between each subsequent software package. The parsing process is a tedious and time-consuming process, highly susceptible to human error. The outcome of this appendix is to integrate some of the most popular circuit design tools into the XIC environment. We present an overall view of the core functionality of the XIC integration tools.

A.2 Design Overview

The XIC design environment extends to interface with both InductEx and JoSIM. Figure A.1 shows the design flow for the XIC integration process. The extension integrates JoSIM into the XIC environment as an additional SPICE based circuit simulation tool. JoSIM simulates the modified electrical netlist extracted from the circuit schematic. The resulting output is converted into a format compatible with XIC and the WRspice plotting functionality. The added functionality allows users to simulate the circuit schematics using both WRspice and JoSIM.

The InductEx integration added an option to extract the inductance from the circuit and back annotate the results without leaving the main window from XIC. The electrical netlist and flattened GDSII is passed to InductEx together with the Layer Definition File (LDF) [90]. The InductEx simulation executes in the background and creates an output file with the results. The output from InductEx is back annotated into XIC, replacing the circuit component values.

The JoSIM and InductEx integration are discussed in more details later in this appendix.

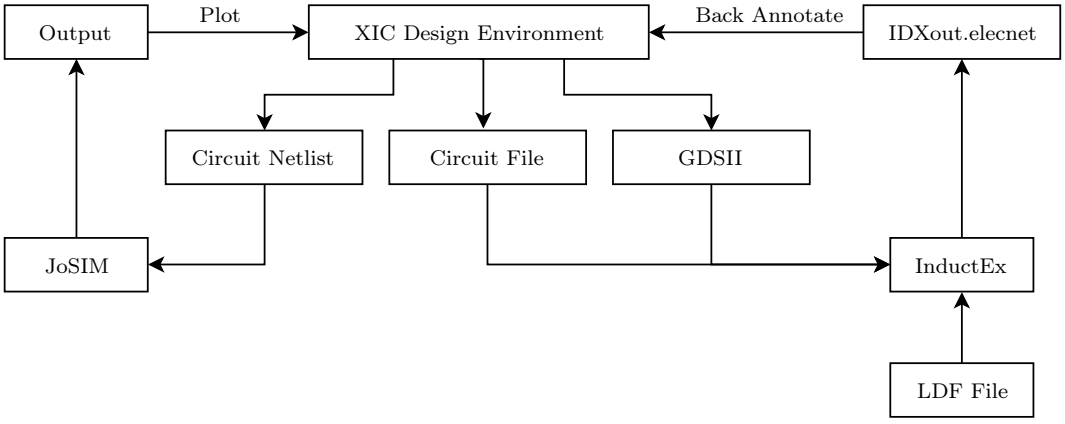


Figure A.1: The Graphical representation of the JoSIM and InductEx integration into the XIC design environment.

A.3 JoSIM Integration

The XIC Layout package makes use of WRspice as the default circuit simulation tool during the design process. WRspice is a multipurpose built-in SPICE engine derived from Berkeley SPICE3 [91]. With the advances made in other SPICE-based circuit simulation tools, it will be of great interest to integrate additional circuit simulation tools for verification purposes and increased performance. The time-consuming process of restructuring the netlist file into a format compatible with the built-in circuit simulators, inhibits the use of external simulation tools.

The circuit simulator selected for integration in the XIC environment was JoSIM. JoSIM is a popular SPICE based circuit simulator widely used by the superconductor circuit design community [92–96]. JoSIM is designed to better manage superconducting circuit elements such as JJs with an increase in simulation speed [97, 98]. The purpose of the JoSIM integration is to add the functionality of JoSIM into XIC without the tedious data extraction.

A.3.1 File Interface

An interface between the circuit schematic in XIC and the circuit simulator JoSIM needs to be created. Figure A.1 shows the desired interface between XIC and JoSIM. A modified SPICE netlist is created from the circuit under test in the XIC environment. The SPICE netlist is a space-separated file

containing the component values, coordinates, interconnects and analysis command. JoSIM simulates the netlist and dumps the results in an output file. XIC processes the output file and plots the results.

A.3.1.1 Netlist Extraction

XIC generates the SPICE netlist from the circuit, excitation conditions and plot command provided in the electrical design environment. The plot function in the current format is incompatible with the JoSIM syntax. The JoSIM documentation requires reformatting the node voltages, the current through the circuit and the phase voltage of the JJs [98]. The netlist function reformats the plot function into a format compatible with JoSIM. The function passes the netlist to JoSIM and simulates in the background. Figure A.2 shows the flow diagram of the netlist process function.

A.3.1.2 JoSIM Output Reformat

JoSIM loads the simulated results into a space-separated output file compatible with the WRspice plotting function. The changes made to the phase and current plotting during the netlist extraction must be reverted. The phase and current plot are reverted into the format used in the WRspice plotting function. For the phase plot, *V(JJ_3rd_Node) V* replaces the original *P(Comp_Name) P* generated by JoSIM. The current plot is rectified by removing the "*CURRENT_*" from the current component name.

A.3.2 Simulation

JoSIM makes use of the *.cir* file generated from a circuit schematic captured in XIC. The SPICE netlist generated from the schematic is restructured and dumped to the current working directory. XICtools searches for JoSIM in the environmental PATH and transfers the restructured SPICE file to the input of JoSIM. JoSIM simulates the SPICE netlist using the analysis command as input. The simulated results are loaded as an output data file in the current working directory. The output data file contains the simulated values of the nodes at each time step.

A.3.3 Plot Function

The plot function identifies the last simulation method used on the current circuit. The results from WRspice are used in the current format. If JoSIM were used, the plot function restructures the output file generated by JoSIM into a format compatible with WRspice. The WRspice plotting engine plots the converted output file with full functionality of the plotting window. The current circuit can be simulated using each simulation method and plotted side by side.

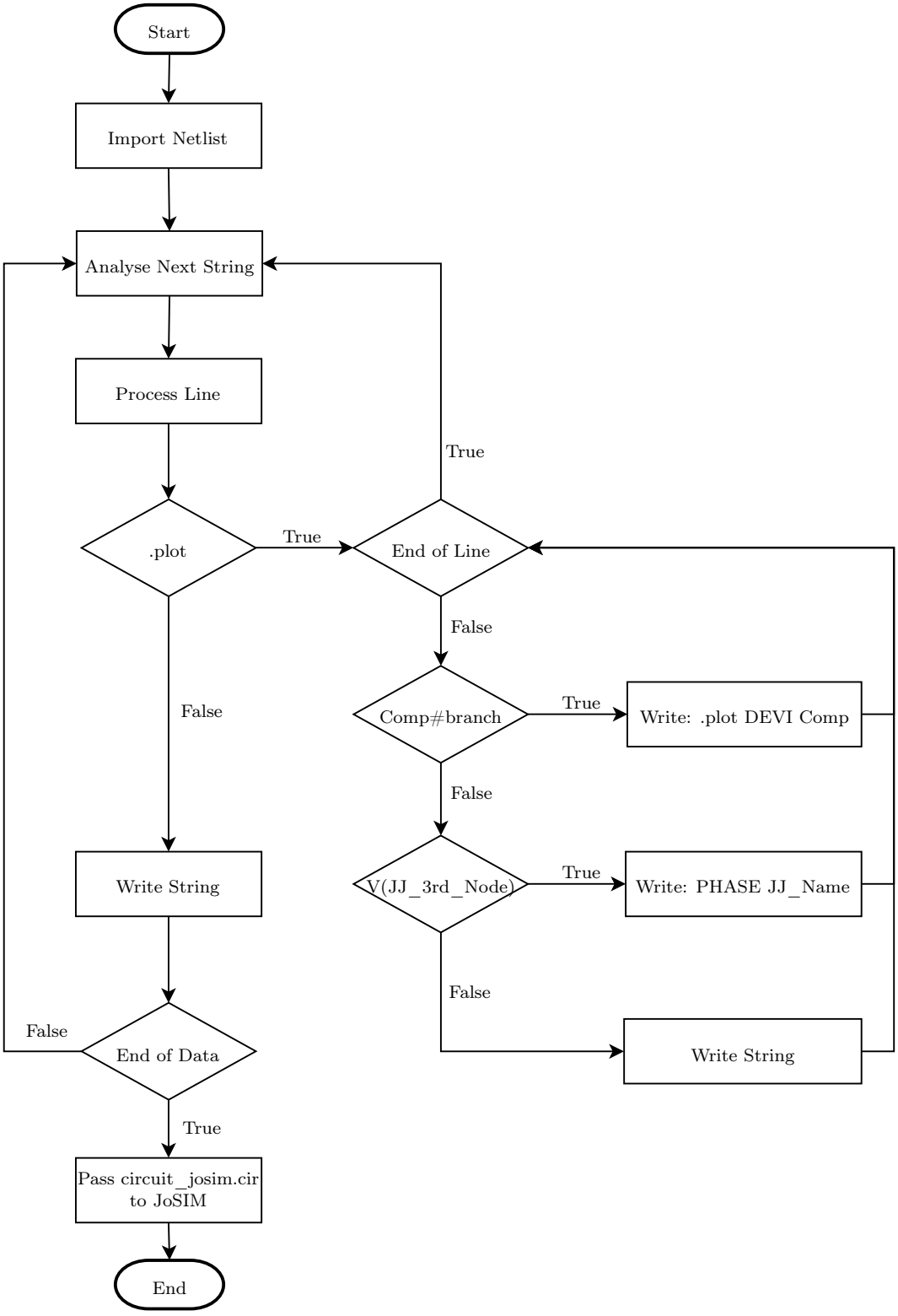


Figure A.2: The flow diagram of the netlist process function between XIC and JoSIM.

A.4 Inductex Integration

The integrated function adds InductEx into the XIC environment to speed up the design process. InductEx extracts the inductance and resistance values from the layout, using the circuit schematic and the GDSII layout. These extracted inductance values are back-annotated into the circuit netlist and the circuit can be simulated again with WRSPICE or JoSIM. The integration function eliminates the need to manually transfer results between XIC and InductEx. The following section explains the file parsing and simulation process in more detail.

A.4.1 File Processing

InductEx expects the circuit netlist and geometry to extract the inductance of the circuit. The circuit netlist consists of a parsed form of the electrical SPICE netlist of XIC. The geometry file contains the physical layout of the circuit. The geometry file extracted by XIC is in the GDSII format. The circuit and physical layout of the circuit is generated separately and combined during the simulation. The cells designed in XIC is extracted and reformatted into a format used by InductEx. InductEx passes the circuit netlist with the extracted parameters to XIC for back annotation. The integration between XIC and InductEx requires an interface that provides a data flow between the software packages. As seen in Figure A.1, the interface is achieved by the creation of four data files.The circuit netlist and the GDSII file are created by XIC. The output netlist, which contains the annotated inductance values, is created by InductEx using the LDF file.

A.4.1.1 Circuit Extraction

The circuit netlist is a modified version of the electrical netlist generated by XIC. The circuit netlist necessitates the refactoring of components, coordinates and values into a format compatible with InductEx. The restructured netlist contains inductors, mutual inductors and the added ports designed for the circuit model. The parsing process removes the rest of the components. The ports from the circuit netlist must be identical to those from the geometry file. The port identification algorithm is responsible for the port placement in the circuit netlist. The algorithm parses the electrical netlist and create the circuit netlist. Figure A.3 is a flow diagram describing the port identification algorithm.

A.4.1.2 GDSII Extraction

The Layout of the circuit schematic is extracted from the physical layer in XIC. The desired port locations are placed in the physical layer of the XIC

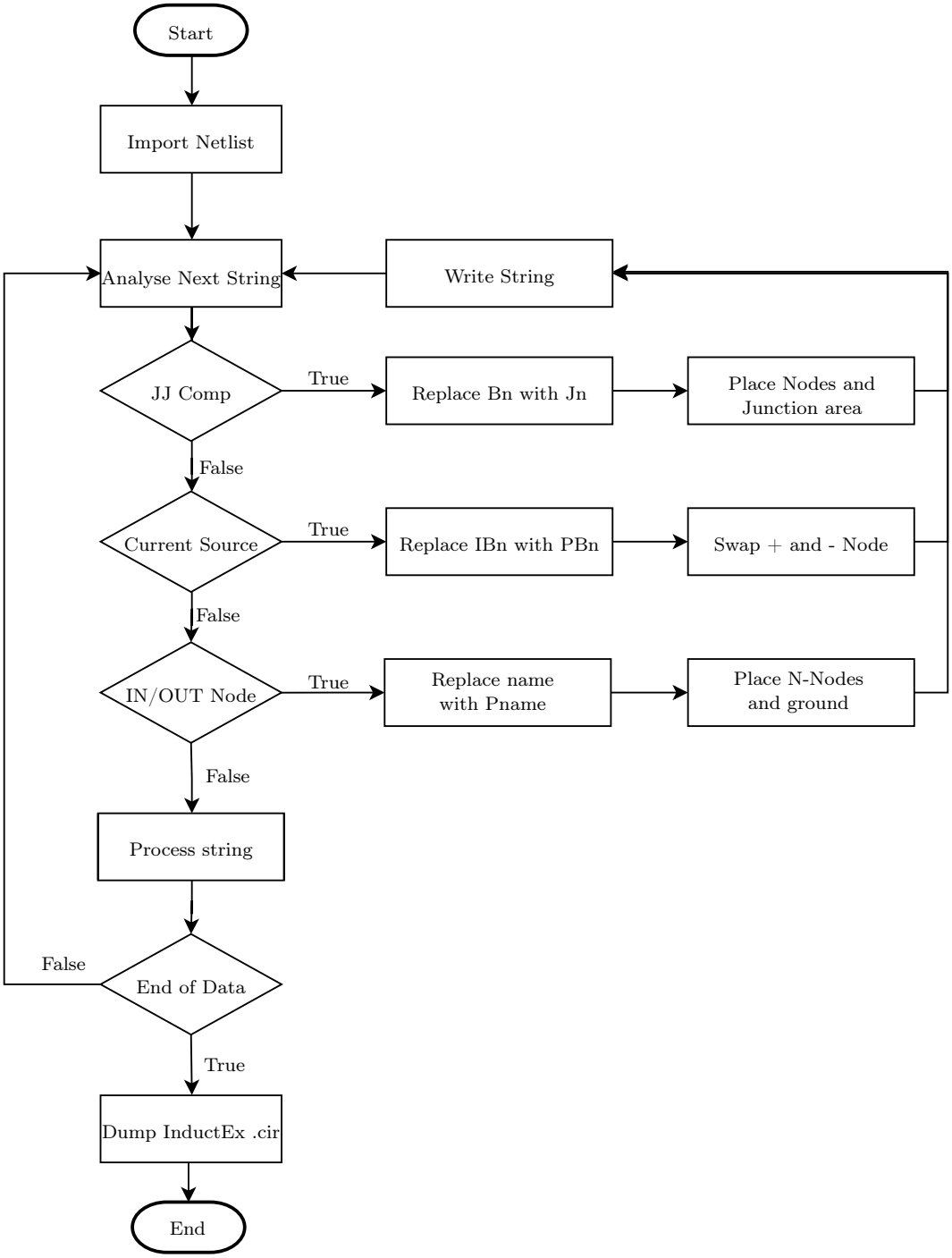


Figure A.3: Port Identification Algorithm flow diagram.

environment during the design process. The ports of the physical layer must match the ports added in the electrical layer of the netlist. The physical layer with the included ports is converted to a GDSII layout file. The GDSII file requires pre-processing before the extracted file is compatible with InductEx. The built-in flattening function in XIC flattens the GDSII file before passing

it to InductEx.

A.4.1.3 Netlist Back Annotation

InductEx outputs a circuit netlist with the extracted impedance values when executed from XIC. InductEx outputs the SPICE based file as IDXout.electnet. The IDXout file consists of the inductors, JJs and mutual inductors that were sent for extraction from the circuit schematic in XIC. The component names of the IDXout file is compared to those in the circuit schematic. If the component names match, the extracted values from IDXout replaces the values of the circuit schematic. The functionality is valid for all component names set by the user.

A.4.2 Simulation

The "run InductEx" command, which is placed in the electrical view of the XIC environment, commences the InductEx simulation process. The "run InductEx" command initiates the file parsing for the circuit netlist and geometry file. XIC creates the extracted files in the current working directory. The LDF file is then selected from a prompt in the command terminal. After that, the numerical engine for the simulation is selected. This is either TetraHenry or FFH (based on FastHenry) depending on the circuit. The LDF, GSDII, circuit netlist and numerical method selected for the simulation are passed to InductEx. The simulation executes in a separate terminal window. Each cell is simulated separately to test the functionality of the cells.

A.4.3 Back Annotation

InductEx generates the solution file with the extracted inductance values for the circuit simulation. The back annotation function restructures the file created by InductEx into a format compatible with XIC. XIC imports the modified file as a circuit netlist into the schematic. The extracted values replace the component values in the circuit schematic of XIC. The JoSIM and InductEx simulation can immediately be performed on the updated circuit schematic.

A.5 Results

The section demonstrates the inductance extraction and verification functionality added to the XIC environment. The integrated functions were tested using RSFQ [99] and AQFP [100–102] cells. The RSFQ cells are tested with JoSIM and WRSpice by using a test bench in XIC. The test bench provides the input for the test cell and processes the output through the sink resistor. Figure A.4 shows the typical test circuit for RSFQ cells. The physical layout

of the RSFQ circuits and Test Bench used for the simulation was designed by Lieze Schindler from Stellenbosch University [89].

The cells used for testing include the Josephson Transmission Line (JTL), OR2T and the AFQP buffer. The actual parameters of the optimized circuit are extracted using InductEx in the XIC environment. The extracted values are back annotated into the circuit schematic of the cell. The cells with the actual inductance values are simulated using JoSIM and WRspice to verify whether the cell still operates as designed. The cell is then passed on for further testing and verification. This includes margin analysis, yield analysis and timing extraction.

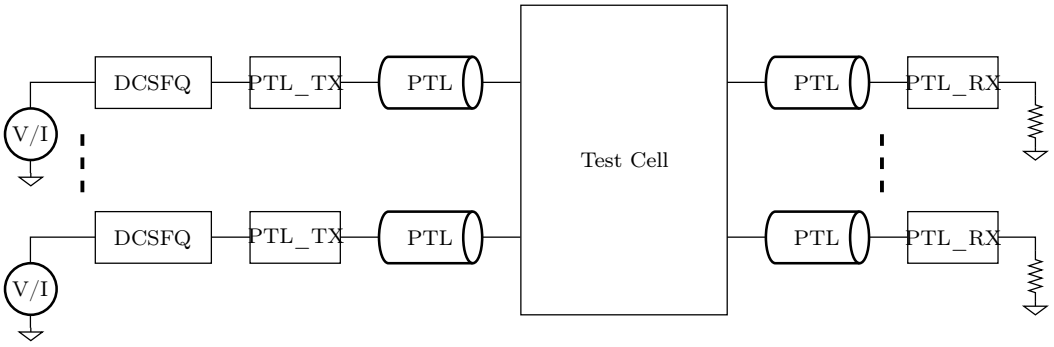


Figure A.4: The test bench circuit diagram for the logic simulator.

A.5.1 Josephson Transmission Line (JTL)

The JTL is a basic RSFQ circuit responsible for SFQ pulse transfer. The SFQ pulse is transferred from one cell to another while matching the cells. The JTL serves as the first example to show the extraction and verification functionality of the integrated tools. Figure A.5 shows the circuit schematic of the optimized JTL before parameter extraction.

A.5.2 Parameter Extraction

The JTL schematic from Figure A.5 must be restructured to include ports before XIC passes the schematic to InductEx for inductance extraction. The port function identifies the JJs, current sources, input and output nodes and replaced them with the corresponding port. Figure A.6 shows the modified circuit schematic displaying the inserted ports into the circuit netlist.

The ports inserted in the schematic must match the ports assigned in the physical layout. If it deviates, the InductEx simulation fails due to mismatched ports. Figure A.7 shows the ports applied to the physical layout of the JTL. XIC passes the modified JTL schematic, generated geometry file and LDF to InductEx.

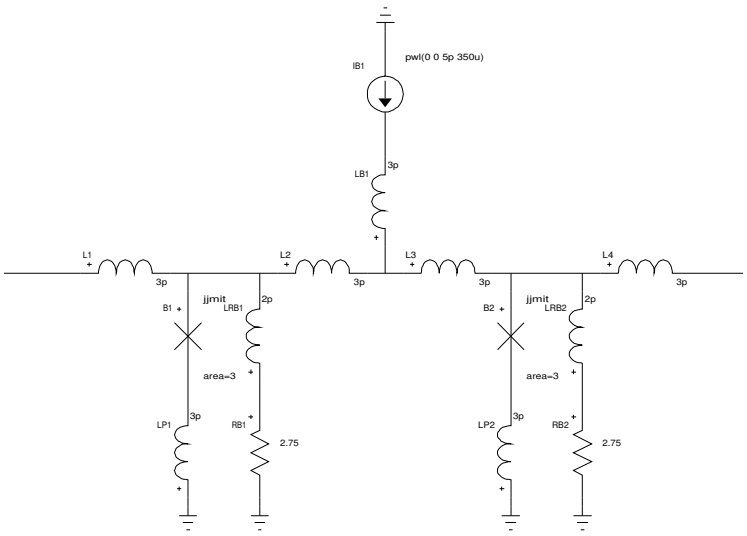


Figure A.5: The optimised JTL circuit schematic with the plot node numbers.

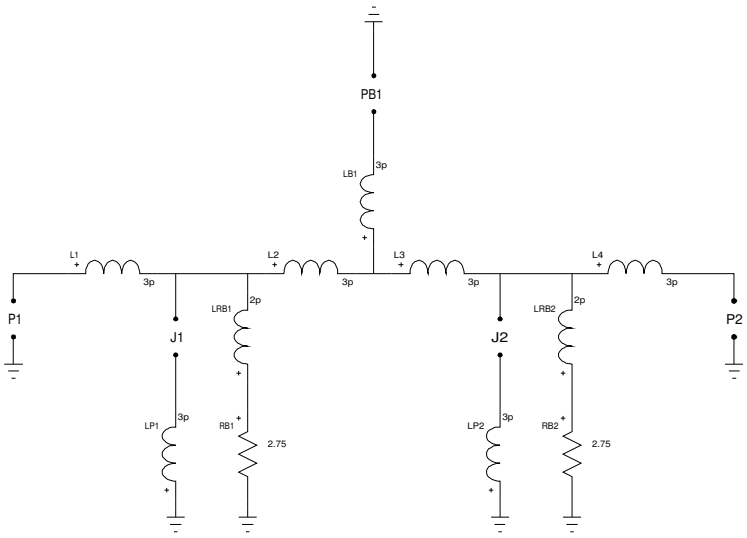


Figure A.6: The modified JTL circuit schematic with included ports.

The TetraHenry numerical field solver engine was used for the InductEx simulation. InductEx creates the circuit netlist with the extracted impedance values into an IDXout.elecnet file. Figure A.8 shows the designed and extracted impedance values for the JTL schematic. The designed and extracted values for the inductors and ports differs slightly. The "import InductEx" command in the extract menu of XIC back annotates the extracted values from the IDXout file to the JTL schematic. Figure A.9 shows the IDXout file generated for the JTL in the current work environment.

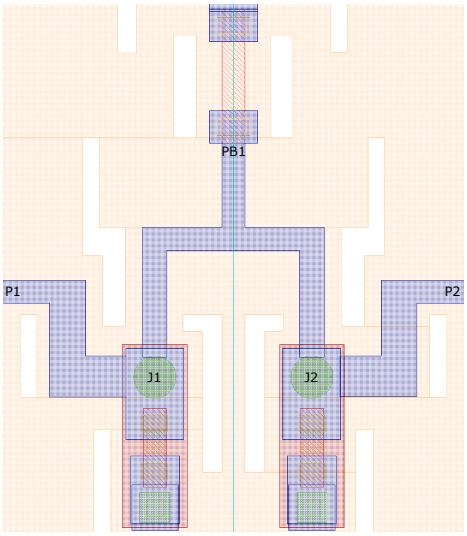


Figure A.7: The JTL circuit schematic physical layout.

Impedance Inductance [pH]		
Name	Design	Extracted
L1	2.00000	1.97836
L2	2.00000	1.99742
L3	2.00000	1.98483
L4	2.00000	1.97613
LB1	0.20000	0.99655
LP1	0.20000	0.45799
LP2	0.20000	0.457716
Ports	Design	Extracted
J1	250.00	260.54
J2	250.00	260.54

Figure A.8: The designed and extracted values for JTL circuit.

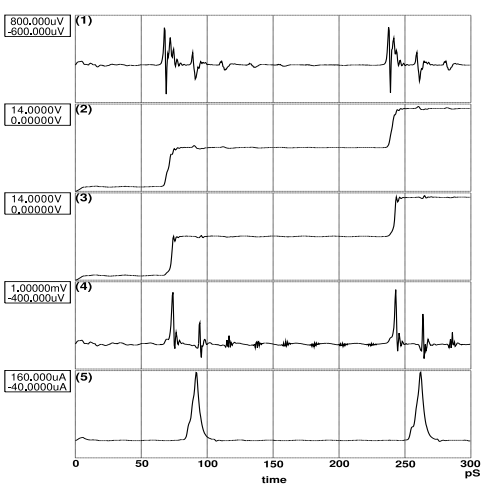
A.5.3 Verification

The circuit is verified using JoSIM and WRspice after the back annotation of the extracted values. The SPICE verification serves as a tool to ensure the cell still perform as designed. The JTL circuit is inserted into the test bench for the SPICE simulations shown in Figure A.4. The nodes to be plotted must be selected before the simulations. Figure A.5 shows the nodes selected by the plotting function. The nodes selected is the phase voltage of the JJs, the input and output node voltage, and the final output. The simulation runs for both JoSIM and WRspice, plotting the results between methods.

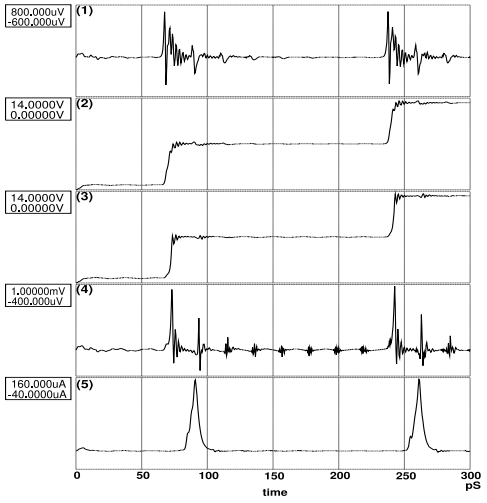
Figure A.10 shows the simulated results for the back annodated JTL cell.

```
* output generated by InductEx
L1 N_P1 3 1.9784E-012
L2 3 6 1.9974E-012
L3 6 4 1.9848E-012
L4 4 N_P2 1.9761E-012
LB1 5 6 9.9655E-013
LP1 7 0 4.5799E-013
LP2 9 0 4.5772E-013
B1 3 7 10000 JMITLL AREA=2.60545E+000
B2 4 9 10000 JMITLL AREA=2.60545E+000
.end
```

Figure A.9: The IDXout SPICE netlist of the JTL circuit generated from InductEx.



(a) JoSIM output of back annotated JTL.



(b) WRspice output of back annotated JTL.

Figure A.10: The results from the JTL after the InductEx back annotation process.(1) shows the SFQ pulse at the data input, (2) and (3) shows the phase voltage of the JJs, (4) the output inductor and (5) the output at the sink resistor. The label numbers match the labels from figure A.5.

Figure A.10a and Figure A.10b both show the circuit functions correctly with the current component values. The circuit operates as expected thus verifying the operation of the circuit after the back annotation of the extracted values.

A.5.4 OR2T

The OR2T cell operates with two logic inputs and a clock input. The output pulse is generated when a SFQ pulse is received from ether logic input before the clock signal is received. The OR2T serves as a larger, more complex example to show the robustness of the integrated tools. Figure A.11 shows the

circuit schematic of the OR2T with the plot nodes used for the JoSIM and WRspice simulation.

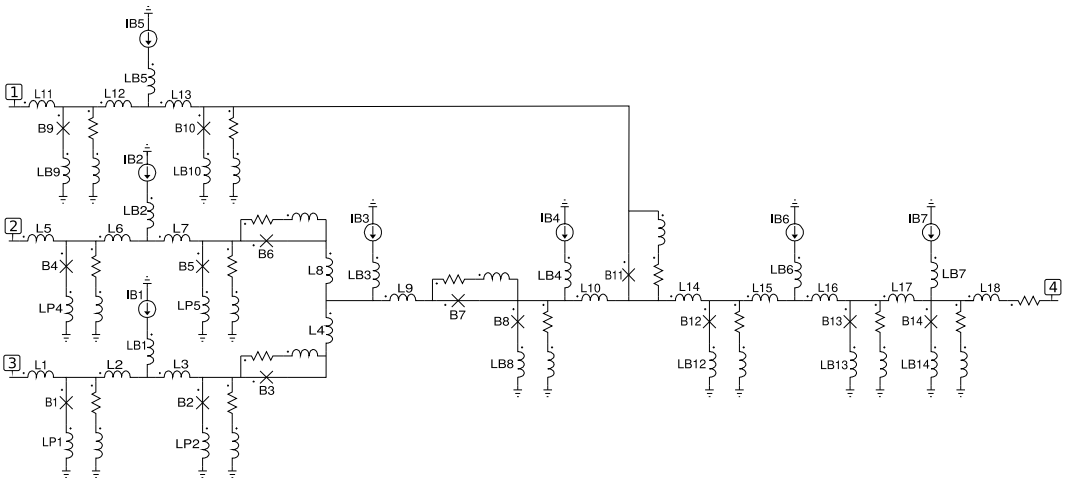


Figure A.11: The OR2T circuit schematic with the plot node numbers.

A.5.5 Parameter Extraction

The first step is to extract the inductance from the optimized OR2T circuit. The inductance extraction is performed by InductEx integrated into the XIC environment. The OR2T schematic needs to have ports added before the netlist is passed to InductEx. The Port Identification Algorithm adds the ports to the schematic. Figure A.12 shows the resulting modified circuit schematic with the ports added to the circuit schematic.

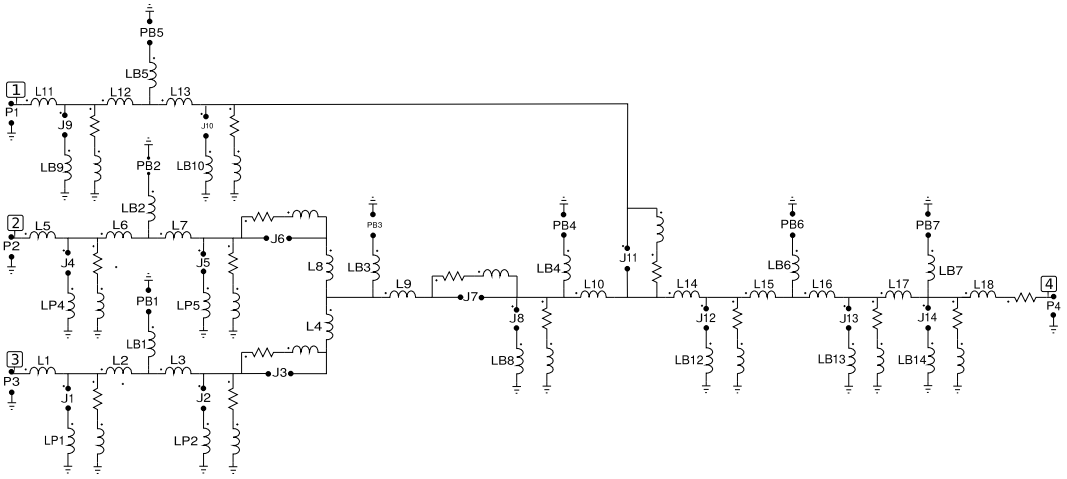


Figure A.12: The modified OR2T circuit schematic with included ports.

The ports inserted in the schematic must match the ports assigned in the physical layout. If a deviation is detected, the InductEx simulation fails due to mismatched ports. Figure A.13 shows the ports applied to the physical layout. XIC passes the modified circuit schematic, generated geometry file and LDF to InductEx.

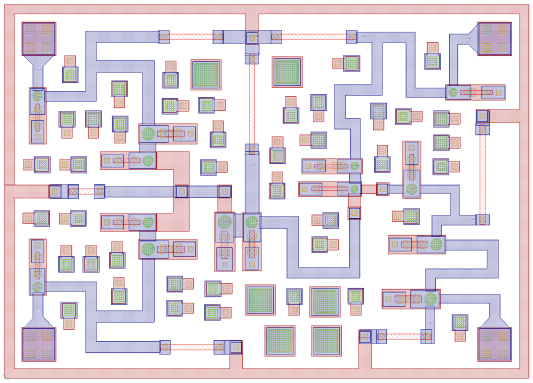


Figure A.13: The OR2T circuit schematic physical layout.

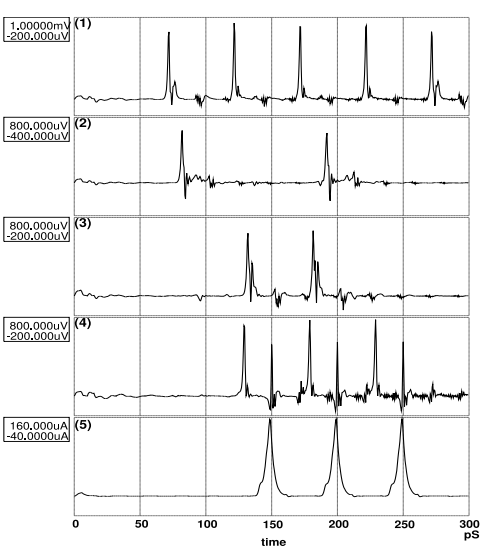
The TetraHenry numerical field solver engine was used for the InductEx simulation. InductEx creates the circuit netlist with the extracted impedance values into an IDXout.elecnet file. Figure B.1, in Appendix B, shows the designed and extracted impedance values for the OR2T cell. The "import InductEx" command in the extract menu of XIC back annotates the extracted values from the IDXout file to the OR2T schematic. Figure B.2, in Appendix B, shows the IDXout file generated for the OR2T cell in the current work environment.

A.5.6 Verification

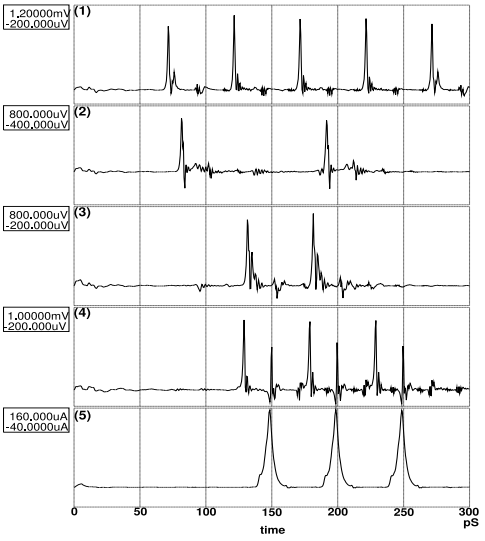
The circuit is verified using JoSIM and WRspice after the back annotation of the extracted values. The SPICE verification serves as a tool to ensure the cell still perform as designed. The OR2T cell is inserted into the test bench shown in Figure A.4 for the SPICE simulations. Figure A.11 shows the clock input, the two logic inputs and the output selected for plotting.

Figure A.14 shows the simulated results for the optimised back annotated OR2T cell. Figure A.14a and Figure A.14b both shows the circuit functions correctly with the extracted component values. The output graph on line 4, generates a pulse for every logic input received during the previous clock period.

.



(a) JoSIM output after InductEx back annotation.



(b) WRspice output after InductEx back annotation.

Figure A.14: The simulation results from the RSFQ OR2T after the InductEx back annotation process. (1) shows the clock pulse at the clock input, (2) and (3) shows the input pulse at the logic inputs, (4) at the output inductor and (5) is the output at the sink resistor. The label numbers match the labels from Figure A.11.

.

A.5.7 AQFP Buffer

The AQFP buffer circuit serves as an example to test the mutual inductance extraction. The AQFP buffer cell was created using the high-speed standard process (HSTP) [101–103]. Figure A.15 shows the circuit schematic of the buffer with optimised component values.

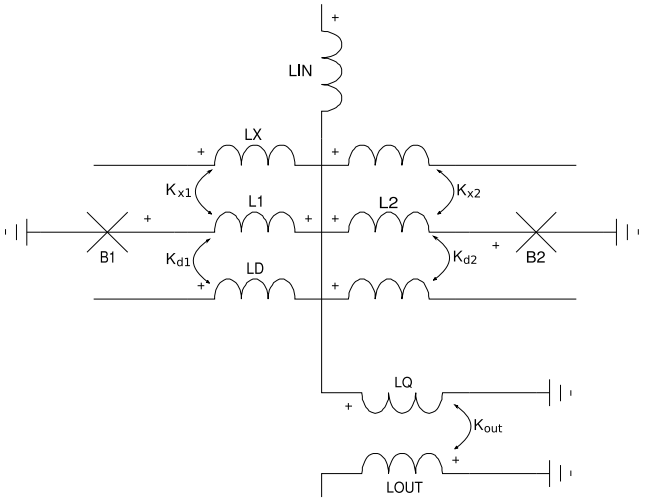


Figure A.15: The AQFP buffer with the mutual inductance shown on the circuit.

A.5.8 Parameter Extraction

The first step performs inductance extraction on the AQFP Buffer circuit. The inductance, coupling factor and junction area are extracted from InductEx. The inductance extraction is performed by InductEx integrated into the XIC environment. Ports are added to each input, output and JJ of the Buffer. Figure A.16 shows the resulting modified circuit schematic with the ports added to the circuit schematic. The mutual inductance is added to the netlist file during the port processing function.

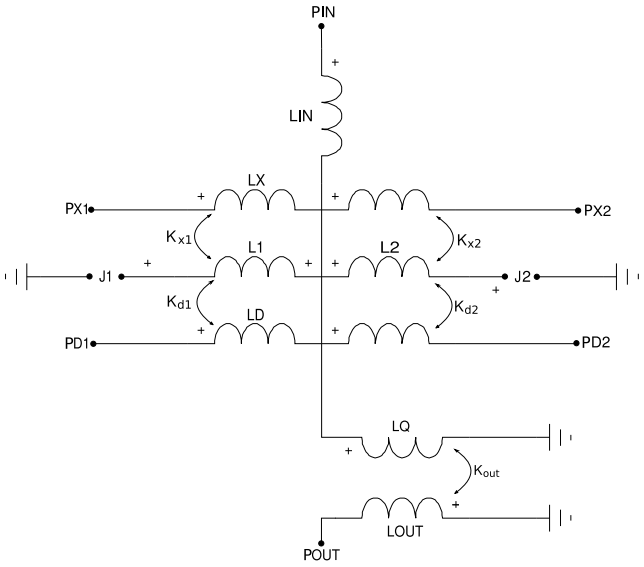


Figure A.16: The modified AQFP buffer circuit schematic with included ports.

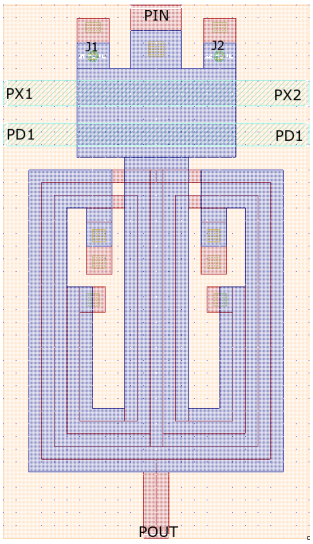


Figure A.17: The AQFP buffer circuit schematic physical layout.

The geometry file is generated from the physical layer of the circuit schematic. Figure A.18 shows the matching ports added to the physical layer before the GDSII extraction. The ports are added manually and must correspond with the ports in the circuit netlist. The modified circuit netlist is passed with the geometry file and the LDF to InductEx.

Figure A.18 shows the original and extracted values for the Buffer circuit. The TetraHenry numerical field solver engine was selected for the InductEx simulation. The extracted inductance values of L1 and L2 is almost identical, which is to be expected for the AQFP buffer design. The AQFP buffer was designed to be symmetrical to reduce unwanted coupling in the circuit [104]. The extracted values are passed to XIC for back annotation.

	Impedance	Inductance [pH]
Name	Design	Extracted
L1	1.35000	1.27542
L2	1.35000	1.27291
LD	8.00000	7.92739
LIN	0.80900	0.80290
LOUT	28.9000	25.1701
LQ	7.68000	6.14888
LX	7.00000	7.44024

	Coupling factor
Name	(k)
Kxd	+0.1978
Kd2	+0.1021
Kd1	-0.1022
Kx2	+0.1680
Kx1	-0.1678
Kout	-0.3841

Ports	Design	Extracted
J1	50.0000	55.3136
J2	50.0000	55.3136

Figure A.18: The designed and extracted impedance values for the AQFP Buffer circuit.

A.6 Conclusion

The tools were successfully able to simulate and verify both RSFQ and AQFP circuit examples. The all-in-one design tool was created utilizing XIC for the circuit design environment, InductEx for the inductance extraction and JoSIM for the circuit simulation. The all-in-one circuit design tool was used to design the circuit, perform circuit simulation on the circuit, extract the inductance from the circuit and layout, back annotate the results into the circuit and perform the circuit simulation again. The integration simplifies the superconductor IC fabrication by having the design and test environment in one package. This environment assists with eliminating the time consuming process of file parsing and transfer between the software solutions. Future work will focus on incorporating support for timing verification. This will be achieved by integrating TimeEx [105] into the XIC environment.

Appendix B

OR2T Simulation

Impedance	Inductance [H]		AbsDiff	PercDiff
Name	Design	Extracted	(L only)	(L only)
L1	8.1302E−13	7.63345E−13	−4.9675E−14	−6.1099%
L2	2.23491E−12	2.05556E−12	−1.7935E−13	−8.0251%
L3	2.85966E−12	2.62654E−12	−2.3312E−13	−8.1521%
L4	1.47671E−12	1.45949E−12	−1.7224E−14	−1.1664%
L5	8.7E−13	8.15418E−13	−5.4582E−14	−6.2738%
L6	2.19475E−12	2.01627E−12	−1.7848E−13	−8.1319%
L7	2.83405E−12	2.6159E−12	−2.1815E−13	−7.6974%
L8	1.4754E−12	1.46392E−12	−1.1482E−14	−0.77824%
L9	2.06452E−12	1.95212E−12	−1.124E−13	−5.4443%
L10	6.43002E−12	5.94014E−12	−4.8988E−13	−7.6187%
L11	2.0718E−12	1.95173E−12	−1.2007E−13	−5.7956%
L12	3.42664E−12	3.15458E−12	−2.7206E−13	−7.9395%
L13	4.10775E−12	3.79651E−12	−3.1124E−13	−7.5768%
L14	1.87923E−12	1.83299E−12	−4.6237E−14	−2.4604%
L15	2.32359E−12	2.0999E−12	−2.2369E−13	−9.627%
L16	1.0271E−12	9.15359E−13	−1.1174E−13	−10.879%
L17	4.62727E−12	4.26165E−12	−3.6562E−13	−7.9014%
L18	2.62304E−12	2.47131E−12	−1.5173E−13	−5.7845%
LB1	2.66854E−12	2.50207E−12	−1.6647E−13	−6.2383%
LB2	2.73559E−12	2.58076E−12	−1.5483E−13	−5.6597%
LB3	2.23242E−12	2.12052E−12	−1.119E−13	−5.0126%
LB4	1.34053E−12	1.26683E−12	−7.3703E−14	−5.498%
LB5	7.7237E−13	7.31281E−13	−4.1089E−14	−5.3198%
LB6	6.5544E−13	6.35596E−13	−1.9844E−14	−3.0276%
LB7	7.6387E−13	7.10807E−13	−5.3063E−14	−6.9466%
LP1	4.9115E−13	4.789E−13	−1.225E−14	−2.4942%
LP2	4.6807E−13	4.67507E−13	−5.6284E−16	−0.12025%
LP4	4.9417E−13	4.81995E−13	−1.2175E−14	−2.4637%
LP5	4.5955E−13	4.60356E−13	+8.0626E−16	+0.17544%
LP8	2E−13	5.03398E−13	+3.034E−13	+151.7%
LP9	2E−13	5.72795E−13	+3.728E−13	+186.4%
LP10	2E−13	7.3845E−13	+5.3845E−13	+269.23%
LP12	2E−13	4.54795E−13	+2.5479E−13	+127.4%
LP13	2E−13	4.88888E−13	+2.8889E−13	+144.44%
LP14	2E−13	4.55832E−13	+2.5583E−13	+127.92%

Figure B.1: The impedance extraction from the OR2T circuit.

Ports	Design	Extracted	AbsDiff	PercDiff
J1	125.73	125.73		
J2	205.1	205.1		
J3	139.73	139.73		
J4	125.73	125.73		
J5	205.1	205.1		
J6	139.73	139.73		
J7	230.7	230.7		
J8	181.78	181.78		
J9	88.638	88.637		
J10	83.195	83.195		
J11	71.138	71.137		
J12	148.64	148.64		
J13	171.14	171.14		
J14	199.81	199.88		

Figure B.2: The JJ area extraction from InductEx.

```
* output generated by InductEx
L1 IN_2 19 7.6335E-013
L2 19 50 2.0556E-012
L3 50 17 2.6265E-012
L4 12 18 1.4595E-012
L5 IN_1 9 8.1542E-013
L6 9 27 2.0163E-012
L7 27 7 2.6159E-012
L8 8 12 1.4639E-012
L9 12 10 1.9521E-012
L10 11 13 5.9401E-012
L11 IN_CLK 6 1.9517E-012
L12 6 22 3.1546E-012
L13 22 5 3.7965E-012
L14 13 14 1.8330E-012
L15 14 38 2.0999E-012
L16 38 15 9.1536E-013
L17 15 16 4.2616E-012
L18 16 41 2.4713E-012
LB1 40 50 2.5021E-012
LB2 26 27 2.5808E-012
LB3 28 12 2.1205E-012
LB4 29 11 1.2668E-012
LB5 22 25 7.3128E-013
LB6 31 38 6.3560E-013
LB7 32 16 7.1081E-013
LP1 52 0 4.7890E-013
LP2 54 0 4.6751E-013
LP4 34 0 4.8200E-013
LP5 36 0 4.6036E-013
LP8 42 0 5.0340E-013
LP9 23 0 5.7280E-013
LP10 20 0 7.3845E-013
LP12 44 0 4.5479E-013
LP13 46 0 4.8889E-013
LP14 48 0 4.5583E-013
B1 19 52 10000 JMITLL AREA=1.25727E+000
B2 17 54 10000 JMITLL AREA=2.05100E+000
B3 17 18 10000 JMITLL AREA=1.39730E+000
B4 9 34 10000 JMITLL AREA=1.25727E+000
B5 7 36 10000 JMITLL AREA=2.05100E+000
B6 7 8 10000 JMITLL AREA=1.39730E+000
B7 10 11 10000 JMITLL AREA=2.30697E+000
B8 11 42 10000 JMITLL AREA=1.81782E+000
B9 6 23 10000 JMITLL AREA=8.86375E-001
B10 5 20 10000 JMITLL AREA=8.31950E-001
B11 5 13 10000 JMITLL AREA=7.11375E-001
B12 14 44 10000 JMITLL AREA=1.48640E+000
B13 15 46 10000 JMITLL AREA=1.71142E+000
B14 16 48 10000 JMITLL AREA=1.99805E+000
.end
```

Figure B.3: The IDXout.elecnet output for the OR2T circuit.

Appendix C

XIC Manual

JoSIM and InductEx Integration into XIC

Quick Start Guide

Version SU1.2

Bernard H. Venter

Stellenbosch University
South Africa
December 10, 2019

Contents

1 Introduction

The JoSIM and InductEx integration was developed under IARPA contract SuperTools(via the U.S. Army Research Office grant W911NF-17-1-0120). The XIC integration is split into two components, JoSIM and InductEx integration. This adds both JoSIM and InductEx circuit simulation functionality to XIC.

The JoSIM integration for XIC enables users to simulate a circuit with the JoSIM circuit simulator as well as the WRspice circuit simulator. JoSIM uses the .cir file generated from a schematic captured with XIC to simulate the circuit. JoSIM then outputs a file that can be used by the plot function of WRspice. The output of both JoSIM and WRSpice can be plotted together to allow for easy comparison between the two simulation methods.

The InductEx integration for XIC enables users to extract parameter values for a circuit layout using InductEx from the XIC layout environment. The the physical and electrical layer is converted into a format used by InductEx. The whole process executes in a separate terminal window.

2 Prerequisites

InductEx and JoSIM will be required to make use of the added functionality added to XIC. The install locations of each is indicated below:

2.1 JoSIM Install Location

XIC will search for JoSIM in the environment path as "josim".If the environment PATH is not set, the JoSIM executable ("josim") will be searched for in *usr/local/bin* as default. JoSIM is available at *https://github.com/JoeyDelp/JoSIM*

2.2 InductEx install Location

XIC will search for InductEx in the environment path as "inductex". If the environment PATH is not set, the InductEx executable will be searched for in the recommended default location: *usr/local/bin* . InductEx is available at *www.inductex.info*

3 Installing XIC

The XIC version with the integrated JoSIM and InductEx functionality can be installed using the precompiled CentOS 7 packages or from source.

3.1 CentOS 7 Packages

The precompiled packages for CentOS 7 are available on Github. Download the folder containing all the precompiled packages and run the installer as follows:

$ sudo ./wr_install all

Thereafter, XIC will be executable using the following command:

$ /usr/local/xictools/xic/bin/xic

3.2 Source Code

The source code is available at: **https://github.com/bernardventer/xictools.git** Follow the instructions on the **README** page for installation instructions.

4 Example Instructions

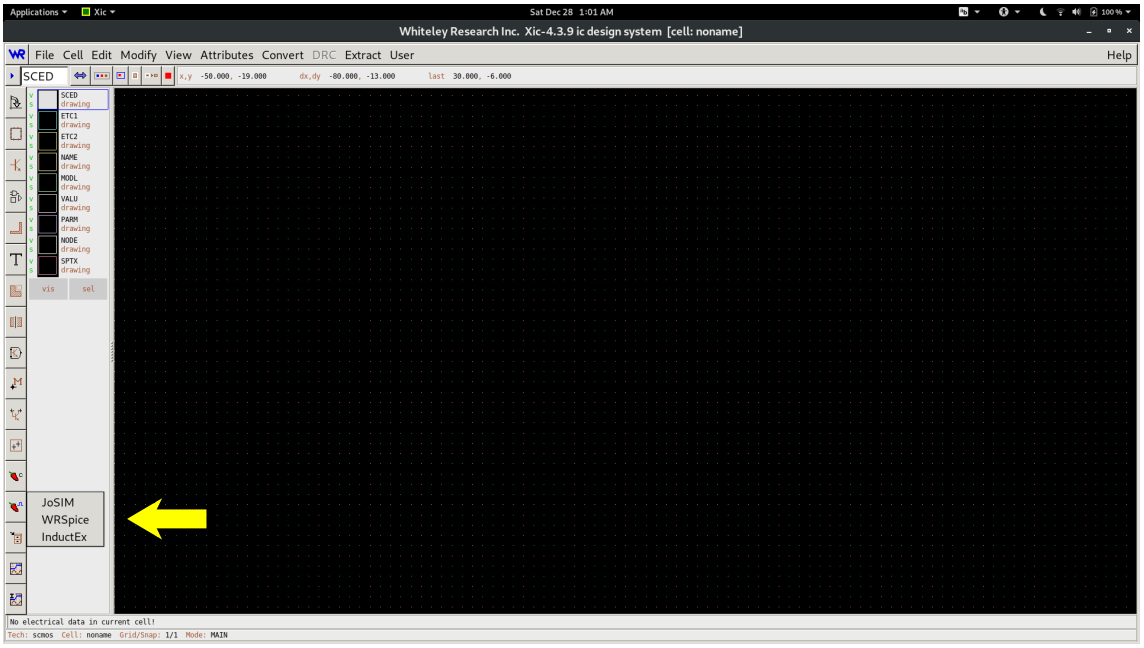


Figure 1: The Run command visible in the Electrical View of XIC. JoSIM, WRspice and InductEx is executed from the Run command.

4.1 JoSIM

1. Open XIC and select File → Open → New.
2. From the Pop-up Window, change the current working directory to the folder containing the circuits (File → new CWD).
3. Load the circuit to be simulated (eg .gds).
4. Switch to the Electrical view by clicking on View → Electrical.
5. **It is recommended to use the WRspice method first**. Run WRspice using using run → WRspice on the side menu and follow the instructions. Figure 1 shows the Run command.
6. Plot the results using plot command and select the nodes to be plotted.
7. Keep the Plot window open for comparison.
8. Run JoSIM using run → JoSIM on the side menu.
9. Enter the analysis command that will be used to simulate the circuit in JoSIM in the prompt box below and press enter.
10. If the JoSIM executable is not in the default location, the file has to be selected through the pop-up file select.

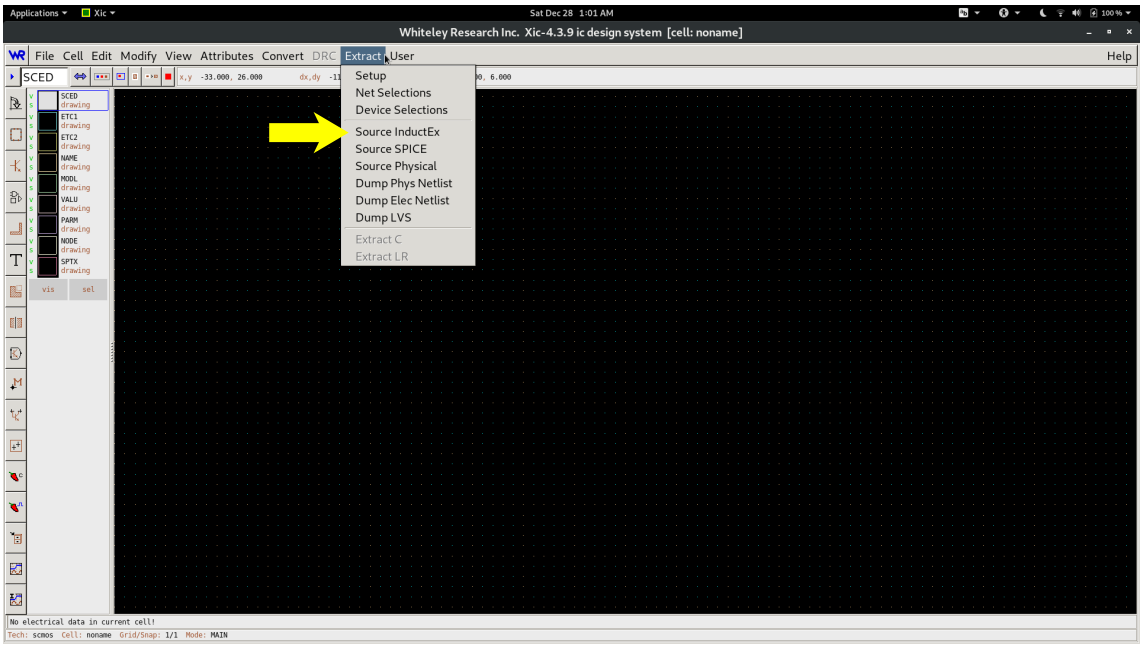


Figure 2: The values extracted from InductEx is back annotated to the XIC environment using the Extract → Source InductEx.

11. Plot the output as for WRspice.

4.2 InductEx

1. Open XIC and select File → Open → New.
2. From the Pop-up Window, change the current working directory to the folder containing the circuits (File → new CWD).
3. Load the circuit to be simulated (eg .gds).
4. Switch to the Electrical view by clicking on View → Electrical.
5. Run InductEx using run → InductEx on the side menu. Figure 1 shows the Run command.
6. Enter the analysis command that will be used to simulate the circuit in InductEx in the prompt box below and press enter.
7. Select the LDF file that will be used for the circuit in the file select menu.
8. Choose which numerical engine to make use of. Type f for FFH and t for TH.
9. The InductEx simulation of the circuit is executed in the terminal window.
10. The output of the simulation can be found in the current working directory of XIC.

4.3 Back Annotate

1. InductEx outputs the extracted impedance values into a circuit netlist file (**IDXout.elecnet**).
2. The extracted impedance values are back annotated by selecting Extract → Source InductEx. Figure 2 shows the InductEx back annotated command.
3. The component values are replaced with the extracted values.
4. The circuit with the updated values are simulated using JoSIM and WRspice.

4.4 Plotting Nodes

Select the desired nodes first using the plot command **This must be done before using the JoSIM command.** This allows the desired nodal values to be loaded into JoSIM for analysis. The nodal values can then be plotted using the plot command again after JoSIM was executed.

5 Output Data

5.1 JoSIM

The data output generated by JoSIM will be dumped in the current working directory. The output data file contains the values of the nodes at each time step. The output file is generated in a format that can be used by the plotting function used by XIC.

5.2 InductEx

The solution files generated by InductEx is found in the current working directory as specified by the user. The extracted inductance from the circuits is outputted as a circuit netlist file called IDXout.elecnet. The IDXout.elecnet file contains the extracted values compatible with XIC.

Appendices

A Recommended Design Rules

The recommended design rules for the electrical and physical layer is given below. These rules ensure the correct ports are generated for the InductEx input file.

A.1 Port Names

1. The Ports for the input and output nodes may have any name **but must start with a P**. The ports will match the names given to the nodes. **Example:** node (internal): v(PIN) will match the input Port PIN in the physical region and node (internal): v(PX1) will match to the Port PX1 in the physical region.
2. **PB** label for the port replacing the current source.
3. **PR** label for the port replacing the resistor connected to ground.

A.2 Naming Convention

1. Rename the JJ to the **B** component label to match that of the port assigned to it. The same as seen on the circuit diagram designed. Ex B1 must be used for J1 port, B10 for J10 etc.
2. Use the **IB** label for the current sources and match the numbering to that of the PB label. Ex. IB3 for port PB3 etc.
3. **RP** is used for the resistors connected to ground that will become ports. Keep the numbering constant. EX RP1 for PR1
4. **RB** for JJ resistor
5. **LRB** for JJ inductor
6. **LP** for inductor connecting the - JJ port to ground.
7. **L** for regular inductors
8. **LB** for current source inductors.

Appendix D

ISEC 2019:Initial Numerical Simulation of the Thermodynamic Behaviour of a Superconducting Circuit

Initial Numerical Simulation of the Thermodynamic Behaviour of a Superconducting Circuit

Bernard H. Venter
Department of Electrical and Electronic Engineering
Stellenbosch University
Stellenbosch, South Africa
17810485@sun.ac.za

Coenrad J. Fourie
Department of Electrical and Electronic Engineering
Stellenbosch University
Stellenbosch, South Africa
coenrad@sun.ac.za

***Abstract*—Localized heating has the potential to create undesired effects in the operation of the superconducting circuits, such as thermal noise and its influence on the SFQ pulse. Left unchecked, it could form into heat zones that could destroy the superconductivity in the circuit. Heat zones only become apparent during the testing phase after manufacture. This process wastes time and materials on a problem that could have been prevented. It is, therefore, crucial to provide a method to simulate the heat propagation before manufacture. We investigate a method to simulate the heat generated by a superconducting circuit during the design process. It will help circuit designers see potential failures beforehand caused by trapped heat zones. The algorithm takes in an object generated by FEniCS as an input and a basis for the heat conduction calculation. The heat conduction is calculated by making use of the electron conduction and lattice vibrations of the material under investigation.**

***Index Terms*—Thermal conductivity, Circuit simulation, Superconductor electronics, Superconducting integrated circuits**

I. Introduction

With the ever-improving fabrication processes for superconducting electronics, the number of superconducting layers and component density in a wafer increased significantly [1], [2]. The smaller form factor with an increased number of layers creates a potential problem of trapped heat between the layers. The localized heating between the layers may increase past the transitional temperature (Tc) and break superconductivity. This could lead to future cooling problems as the heat is generated faster than it is dissipated in more complex circuits.

We investigate the effect of the total conductivity through a thin-film niobium strip and the resulting heat propagation. The thermal conductivity equation calculated by Koechlin and Bonin [3] was implemented with the general heat conduction partial differentiation equation (PDE) to calculate the temperature at each node [4]. The General heat PDE cannot be solved using analytical methods, the real solution to the PDE is approximate by making use of the numerical method. We approximate the heat propagation through the circuit using the Finite Element Method (FEM) [5]. The calculation is iterated until the approximate relative error of the simulated temperature is $< 1\%$.

The research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office grant W911NF-17-1-0120.

II. Method/Model

A. Heat Conduction Equation

We model the heat conduction through a rectangular two-dimensional object with the x and y-direction propagated along the horizontal and vertical direction respectfully. The model assumes the length in the x-direction is finite and the width in the y-direction is infinite. This approach was selected to only focus on the influence of the total conductivity making use of Dirichlet boundaries in the x-direction. The temperature over object is calculates using the heat conduction equation:

$$p(T)c_p(T)\frac{dT}{dt} = \vec{\nabla}\cdot(K(T)\vec{\nabla}T) + Q. \tag{1}$$

Where K is the thermal conductivity of the material, Q is the rate of energy generated per unit volume, c_p is the heat capacity of the medium, p is the density of the material and T is temperature [4]. For the purpose of this paper, we make use of the steady state case of (1) stated as:

$$\vec{\nabla}\cdot(K(T)\vec{\nabla}T) = 0. \tag{2}$$

The steady state assumes no heat is generated in the circuit and the temperature derivative is zero. We use the steady state case to investigate the influence of the total conductivity on the heat propagation of the circuit in question.

B. Total Conductivity

The stability of a superconducting circuit is largely influenced by the total thermal conductivity of the material used in the layers. The total conductivity is the combination of the electron and lattice conductivity. The lattice conductivity component is due to the scattering of photons. The thermal conductivity of niobium is nonlinear below Tc due to the vanishing electron heat conduction as the temperature decreases [3], [6]. The thermal conductivity equation for the temperature region lower than Tc and is stated by Koechlin and Bonin [3] as:

$$K_s = R(y)\left[\frac{p(295K)}{L \cdot RRR \cdot T} + aT^2\right]^{-1} + \left[\frac{1}{D \cdot exp(y)T^2} + \frac{1}{BlT^3}\right]^{-1}. \quad (3)$$

Where *R(y)* is the ratio of the thermal conductivity due to electrons in the superconducting region to that in the normal region. *R(y)* scales down the thermal conductivity at the lower temperature range due to the formation of Cooper pairs [6]. *L* is the Lorentz constant, *RRR* is the Residual Resistivity Ratio of the metal, *l* is the phonon mean free path, *a* is the coefficient of momentum exchange with the lattice vibrations, *B* is a constant depending on the mechanism of scattering and the material, *D* is the Debye temperature of the metal and *exp(y)* is the condensation into Cooper pairs term [3].

C. FEM

If we apply equation (3) to (2) we obtain a PDE to calculate the temperature in a niobium strip for T $<$ Tc. The approximate solution to the heat conduction PDE is calculated using FEM [7]. We make use of an open source PDE solver, FEniCS [8], to simulate the problem.

The FEM makes use of the PDE in its variational form. The PDE is reformulated into the variational formulation using a few general steps. Equation (2) is multiplied by a test function, *v*, on both sides. The resulting equation is integrated over the domain of the function, Ω. The test and trial, *v* and *u*, function must both be elements of the new function space of the problem *V* [9]. The variational formulation of the heat conduction PDE is derived as:

$$\int_\Omega K\nabla u \cdot \nabla v dx = \int_{\partial\Omega} K\frac{\partial u}{\partial n} v ds. \quad (4)$$

Where $\partial\Omega$ is the domain boundary and is the combination of all the boundary conditions applied. The Dirichlet and Neumann boundaries are of interest for our simulation. The Dirichlet boundary assumes the temperature is kept constant along the boundary. The Neumann boundary specifies the flux normal to the boundary [10].

D. Conduction Iteration

The initial thermal conductivity is calculated using (3) with a constant temperature of 4.2K. The variational form is then simulated for the initial thermal conductivity to find the steady state temperatures. The steady state temperature is substituted into (3) to calculate the new thermal conductivity. The new thermal conductivity is back annotated into the PDE problem and the temperature is recalculated. The relative approximation error is calculated using the current and previous temperature results as follow:

$$\epsilon_a = \frac{u - uh}{u}. \quad (5)$$

Where ϵ_a is the relative approximation error, u is the present approximation and uh is the previous approximation. The simulation iterates until $|\epsilon_a| < 1\%$.

Fig. 1. Heat propagation through a rectangular circuit with the boundaries in the x-direction kept at 2K and 9K respectfully. The black line represents the median temperature of 5.5K

III. Results

The heat conduction is simulated through a rectangular object to investigate the effect of the temperature-dependent thermal conductivity. The niobium thin-film object is assumed to be infinitely long in the y-direction with a fixed length in the x-direction. Dirichlet boundaries are applied to the left and right boundary of the circuit and Neumann on the rest. The temperature on the left boundary is kept at a constant temperature of 2K and the right boundary is kept constant at a temperature of 9K. Since the width in the y-direction is infinitely long and the x boundaries are known, the Neumann boundaries are zero.

The resulting heat propagation of the heat conduction simulation is shown in Fig. 1. The black bar represents the mean temperature of the two heat sources. For an object with a constant thermal conductivity, the bar would be in the middle with a mean value of 5.5K. It can be noted that the bar in Fig. 1 is close to the left boundary with the temperature at the middle of the circuit equal to 7.35K. The thermal conductivity rises rapidly as the temperature increases resulting in the disproportion temperature in the object.

The Thermal conductivity vs temperature was calculated for niobium with similar material properties as in [3]. With a small increase of temperature, 3K to 6K, the respective Thermal conductivity increased approximately ten-fold, 16.5 to 163 $W/m \cdot K$. The strange thermal properties of niobium below Tc are due to the increase formation of Cooper Pairs as the temperature decreases. Thermal conductivity in a metal is dominated by electron conductivity. The formation of Cooper pairs decreases the number of electrons that contributes to the thermal conductivity [6].

IV. Conclusion

The experiment was performed under the Coldflux project [11]. It acts as a test of concept for simulating the heat propagation through thin-film niobium. This concept will provide the basis for future thermal simulations of superconducting circuits. The experiment simulates the nonlinear heat propagation through a 2D model using the temperature-dependent thermal conductivity. The nonlinear spread of temperature in the circuit shows that heat has the potential to get trapped between the

superconducting layers. The low thermal conductivity of the insulator surrounding the niobium reduces the amount of dissipated trapped heat. If the dissipation of heat is too slow, it may break the superconductivity of the circuit. Thus, reiterating the importance of thermal simulations of superconducting circuits.

The method applied above can be extended to 3D since we are making use of numerical methods to calculate the heat propagation through a circuit. The simulation is only valid for problems with well-defined boundary conditions.

ACKNOWLEDGMENT

The authors thank Paul le Roux and Sasan Razmkhah for very fruitful discussions.

REFERENCES

[1] S. K. Tolpygo, V. Bolkhovsky, R. Rastogi, S. Zarr, A. L. Day, E. Golden, T. J. Weir, A. Wynn, and L. M. Johnson, "Advanced fabrication processes for superconductor electronics: Current status and new developments," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–13, Aug 2019.

[2] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, M. Hidaka, H. Akaike, A. Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa, "New nb multi-layer fabrication process for large-scale sfq circuits," *Physica C: Superconductivity and its applications*, vol. 469, no. 15, pp. 1578–1584, 2009.

[3] F. Koechlin, "Parametrization of the niobium thermal conductivity in the superconducting state," *Superconductor Science and Technology*, vol. 9, no. 6, pp. 453–461, June 1996.

[4] Y. A. Cengel, *Heat transfer : a practical approach*, 2nd ed., ser. Mcgraw-Hill series in Mechanical Engineering. Boston: McGraw-Hill, 2003.

[5] COMSOL, "The finite element method (fem)," Feb 2017. [Online]. Available: https://www.comsol.com/multiphysics/finite-element-method

[6] J. Bardeen, G. Rickayzen, and L. Tewordt, "Theory of the thermal conductivity of superconductors," *Physical Review*, vol. 113, no. 4, p. 982, 1959.

[7] E. L. Wilson and R. E. Nickell, "Application of the finite element method to heat conduction analysis," *Nuclear engineering and design*, vol. 4, no. 3, pp. 276–286, 1966.

[8] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The fenics project version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[9] H. P. Langtangen, A. Logg, and A. Tveito, *Solving PDEs in Python: The FEniCS Tutorial I*. Springer International Publishing, 2016.

[10] H. P. Langtangen and K.-A. Mardal, "Introduction to numerical methods for variational problems," 2016. [Online]. Available: http://hplgit.github.io/fem-book/doc/web/

[11] C. J. Fourie *et al.*, "Coldflux superconducting eda and tcad tools project: Overview and progress," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–7, Aug 2019.

Appendix E

Cold Finger Simulation Output for 3.00mA

```
Katana Mesh Found...

Material        Subdomain ID
Nb              ['2', '3']
Mo              ['1']
SiO2            ['7']
Al              []
Si              ['8']

Boundary Type       Bound ID        Condition (K)
Rad                 4               0.018
Dirichlet           6               4.2

Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 4.947e+03 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 4.981e-03 (tol = 1.000e-05) r (rel
    ) = 1.007e-06 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 7.648e+03 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 7.842e-05 (tol = 1.000e-05) r (rel
    ) = 1.025e-08 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 5.275e+02 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.481e-06 (tol = 1.000e-05) r (rel
    ) = 2.807e-09 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.674e+01 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.465e-06 (tol = 1.000e-05) r (rel
    ) = 8.753e-08 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 3.186e+00 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.457e-06 (tol = 1.000e-05) r (rel
    ) = 4.572e-07 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.736e-01 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.401e-06 (tol = 1.000e-05) r (rel
    ) = 8.070e-06 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
```

```
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.757e-02 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.418e-06 (tol = 1.000e-05) r (rel
    ) = 8.072e-05 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.521e-03 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.376e-06 (tol = 1.000e-05) r (rel
    ) = 9.046e-04 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.091e-04 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton iteration 1: r (abs) = 1.341e-06 (tol = 1.000e-05) r (rel
    ) = 1.229e-02 (tol = 1.000e-05)
  Newton solver finished in 1 iterations and 1 linear solver
    iterations.
Solving nonlinear variational problem.
  Newton iteration 0: r (abs) = 1.170e-05 (tol = 1.000e-05) r (rel
    ) = 1.000e+00 (tol = 1.000e-05)
  Newton solver finished in 0 iterations and 0 linear solver
    iterations.

Solver finished in  10 iteration
```